11-1-2018

Data Sampling and Reasoning: Harnessing Optimization and Machine Learning for Design and System Identification

Pei Cao
University of Connecticut - Storrs, pei.cao@uconn.edu

Data Sampling and Reasoning: Harnessing Optimization and Machine Learning for Design and System Identification

Pei Cao, Ph.D.

University of Connecticut, 2018

The recently rapid advancements in sensing devices and computational power have caused paradigm shift in engineering analyses: data driven and sampling-based approaches play more and more important roles. For example, with sampling-based global optimization techniques, mechanical components can be refined and redesigned; system parameters can be accurately identified. With advancements in data mining and machine learning, underlying input and output relations of complex systems can be developed to make predictions with unseen data or to expedite analytical process. In this dissertation, advanced optimization and machine learning techniques are developed and employed to multi-facet engineering tasks from system design to identification.

First research task concerns a type of configuration designs, where the volume occupied components is to be minimized along with other objectives such as the length of connectivity lines. The formulation of computationally tractable optimization is difficult in practice as the objectives and constraints are usually complex. Moreover, the design optimization problems usually come with demanding constraints that are hard to satisfy, which results in the critical challenge of balancing feasibility with optimality. We develop an enhanced multi-objective simulated annealing approach, MOSA/R, to solve this problem. A versatile and efficient re-seed scheme that allows biased search while avoiding pre-mature convergence is designed in MOSA/R. Some generalization studies of the algorithm have also been carried out. In this second task, we exploit the impedance/admittance change of a piezoelectric transducer bonded to a host structure, aiming at the identification of system damage. To find a small set of solutions for such an under-determined system that indicates the true damage scenario, we cast the damage identification problem

into a multi-objective optimization framework. With damage locations and severities as unknown variables, one objective function is the discrepancy between first-principal model predictions and actual measurements. The sparsity of the unknown variables is chosen as another objective function, deliberately, the l0 norm, because damage occurrence generally affects a small number of elements. A multi-objective algorithm (DIRECT) is devised to facilitate the inverse analysis where the sparsity is further emphasized by sigmoid transformation. As a deterministic technique, this approach yields repeatable and conclusive results. The third task concerns early diagnosis of gear transmission, which is challenging because gear faults occur primarily at microstructure but their effects can only be observed at a system level. The performance of a fault diagnosis system depends on the features extracted and the classifier subsequently applied. Fault-related features are conventionally identified based on domain expertise, which are system-specific. On the other hand, although deep neural networks enjoy adaptive feature extractions and inherent classifications, they require a substantial set of training data. We present a deep convolutional neural network-based transfer learning approach, which not only entertains preprocessing free adaptive feature extractions, but also requires only a small set of training data.

Data Sampling and Reasoning:

Harnessing Optimization and Machine Learning for Design and System Identification

Pei Cao

B.S., Northwestern Polytechnical University, 2011

A Dissertation

Submitted in Partial Fulfillment of the

Requirements for the Degree of

Doctor of Philosophy

at the

University of Connecticut

2018

APPROVAL PAGE

Doctor of Philosophy Dissertation

Data Sampling and Reasoning:

Harnessing Optimization and Machine Learning for Design and System Identification

Presented by

Pei Cao, B.S.

Major Advisor _____

Dr. Jiong Tang

Associate Advisor _____

Dr. Horea Ilies

Associate Advisor _____

Dr. Chengyu Cao

Associate Advisor _____

Dr. Xu Chen

Associate Advisor _____

Dr. Ying Li

University of Connecticut

2018

This work is dedicated to my beloved parents, my wife

and my son for their love and endless support

Acknowledgement

First and foremost, I would like to express my sincere gratitude and respect to my major advisor Professor Jiong Tang for his mentoring, support and encouragement throughout my 6-year graduate career. His working ethics, wisdom and passion about research have always been a great source of inspiration for me. I have learned from him not only the academic skills, but also the perspective of career and life. This is truly a rewarding journey which I will cherish and preserve for the rest of my life.

I also would like to express my gratitude to my associated academic advisors, Professor Horea Ilies, Professor Chengyu Cao, Professor Xu Chen, Professor Ying Li, for serving my advisory committee and providing valuable input and suggestions for my dissertation. Their keen insight, helpful advice and critiques on my research project give a tremendous value to me throughout my entire graduate study.

Working in the Dynamics, Sensing and Controls Laboratory has been a terrific and unforgettable experience. I am very fortunate to have worked with my excellent and supportive colleagues Dr. Kai Zhou, Dr. Edward Diehl, Dr. Qi Shuai, Dr. David Yoo, Dr. Jiawen Xu, Dr. Shengli Zhang, Dr. Yuan Yuan, Shilong Li, Arun Hedge, Matthew Cremins and Yixin Yao. It is enjoyable to work with all of you.

Finally, I would express my greatest gratitude to my parents for their unconditional love and support, my wife Jiayue for always allowing me to be me and giving me hope through all the ups and downs, and also to my lovely son Sidney who has brought a tremendous amount of pride and joy to our life.

Table of Contents

Chapter 1. Introduction

1.1 Background and state of the art

1.1.1 Configuration design and optimization

Configuration design and optimization have been widely studied since the 18th century. In modern computer-integrated manufacturing, configuration optimizations are frequently encountered in aerospace and automotive systems (Grignon and Fadel, 2004), manufacturing facilities and plants (Yang and Hung, 2007; Gonzalez-Cruz and Martinez, 2011; Nageshwaraniyer et al, 2013), and 3-dimensonal laser cutting (Gan et al, 2004) etc. In general, configuration design involves a wide variety of goals and objectives, and densest packing is a good example in terms of the difficulties one may encounter when dealing with such topics. In two-dimensional scenarios, one is given a set of geometries such as rectangles, polyminos or spheres. The goal is to pack these items orthogonally into a single rectangular box of unlimited height which needs to be minimized (Bortfeldt and Wäscher, 2013), or alternatively, to pack a number of circles inside a circumcircle whose radius needs to be minimized (Muller et al, 2009). Three-dimensional problems can be defined in a similar fashion, e.g., given a set of three-dimensional objects of arbitrary geometry and an available space (possibly the space of a container), find a placement for the objects within the space that achieves the design objectives, such that none of the objects interferes (i.e. occupy the same space) while optional spatial and performance constraints on the objects are satisfied. The problem of packing has shown to be NP-complete (Garey and Johnson, 1979), i.e., no optimal algorithm is known running in polynomial time. Therefore, even simple design problems involving spheres, squares or rectangles are known to be difficult problems in the mathematical sense (Grignon and Fadel, 2004). As such, heuristic approaches are required to obtain near-optimal solutions in realistic amount of computational time.

Throughout the years, various global optimization strategies have been tailored to address the configuration optimization problems. Dyckhoff (1990) used branch and bound approach to solve simple rectangular layout problems with a small number of objects. Szykman and Cagan (1995) presented a

Simulated Annealing (SA) based approach to generate configurations. Although theoretically general, their work was limited to simple geometric shapes and restricted rotations. Then the work was extended to configuration problems of arbitrary shapes by considering the spatial constrains between components using the same SA approach (Cagan et al, 1998). Meanwhile, Sachdev et al (1998) proposed a framework to integrate techniques such as Simulated Annealing and Genetic Algorithms (GA) to solve configuration design problems. Grignon and Fadel (2004) applied multi-objective GA towards configuration optimization problems with more complex objectives. Later on, Aiello et al (2006) employed a multi-objective constrained genetic algorithm to solve facility layout problems. A recent review of packing and configuration design methods can be found in (Fadel and Wiecek, 2015). It is worth noting that in previous investigations, the feasibility of the design, i.e., finding a configuration that satisfies constraints, has not been a major challenge. On the other hand, many configuration design problems in modern engineering practices, such as the manifold configuration design problem, are subjected to multiple hard constraints derived from practical requirements, e.g., the distance between certain components is restricted, and neither overlap nor out-of-bound is allowed. In these scenarios, finding a configuration that satisfies these constraints itself may become computationally demanding.

1.1.2 Connective plumbing design

The recently rapid development of additive manufacturing technology has induced paradigm-shifting advancements and opportunities on several fronts. One of such opportunities, which has not been thoroughly investigated, is to minimize the lengths/weights of various networking components, e.g., hydraulic piping, in vehicular, marine, and airborne systems where the reduction of sizes and weights of these systems has significant payoff, as partially discussed in Section 1.1.1. Limited by tooling capability, a connecting pipeline designed for traditional manufacturing consists of only straight line segments and machining accesses, whereas the pipelines design for additive manufacturing could be free-form curves of shorter length. The free-form piping design made possible by additive manufacturing could potentially reduce the overall weight and space occupied by the networking component. More

importantly, additive manufacturing process generally doesn't require any specific new tooling, which could save a lot of time in the process of prototype production and validation.

Mathematically, the problem of length optimization of networking components under given constraints, i.e., locations of connection elements, can be preliminarily casted into a known terrain path-finding framework that is frequently encountered in video games, GPS navigation, and robotics and path planning etc (Rabin, 2002; Sturtevant and Geisberger, 2010; Konar et al, 2013; Rakshit et al, 2013; LaValle, 2006; Cui et al, 2016). Other geometric or physical constraints, such as curvatures, pressure drops, and etc., are successively addressed either manually or through an automated procedure. Known terrain path-finding can be generally divided into two steps (Algfoor et al, 2015): 1) discretizing the continuous routing space, and 2) searching along the graph to find the path minimizing the cost value, which represents the overall length of the path from the starting node to the target node for a non-weighted graph. In this study, without losing generality, square grids are used to discretize the terrain owing to its simplicity and popularity in various applications. A review of different terrain discretizations can be found in (Algfoor et al, 2015). One of the earliest investigations was the Dijkstra's algorithm (1959) where the cost value for the incremental search to the nearest goal is used. The well-known A* algorithm (Hart et al, 1968) made some improvements by adding heuristic cost from the current node to the target node to the evaluation. A* is widely used for solving path-finding problems as it has simple algorithmic structure and is guaranteed mathematically to find a solution. However, the shortest path in gridded graph is not equivalent to the shortest path in continuous space where polynomials along the edge of grids can be replaced by straight lines (Rabin, 2000). A commonly adopted solution to address the optimality issue is to apply post-processing techniques to shorten the obtained grid path at the cost of an increase of computational time (Thorpe and Matthies, 1984; Botea et al, 2004). The post processing techniques usually find a shorter any-angle path compared to A*, but the improvement is limited (Zitzler, 1999) since they directly make adjustment to paths obtained by A* without knowing whether the post-processed path is optimal or not (Zhang and Li, 2007).

Aiming at overcoming the limitations of A*, Field D* (Ferguson and Stentz, 2006), an advanced

version of D* Lite (Koenig and Likhachev, 2002), was proposed which uses the linear interpolation of path costs of vertices to obtain any-angle paths. However, Field D* could result in paths that have unnecessary heading changes and should thus be smoothed further (Nash and Koenig, 2013). Some other approaches, such as Theta* (Daniel et al, 2010; Nash et al, 2007) and Lazy Theta* (Nash et al, 2010), were also proposed which embeds smoothing process into the A* searching to release the constraint on path searching directions (45 or 90 degrees). Theta* finds shorter paths in less time than Filed D* according to literature (Nash et al, 2007). Nevertheless, it is not optimal because the parent of a vertex has to be either a visible neighbor of the vertex or the parent of a visible neighbor, which is not always the case for true shortest paths (Daniel and Nash, 2010; Dang et al, 2015). Figure 1.3 shows the optimal path found by Theta* and the true shortest path, in which the path found by Theta* takes unnecessary detours compared to the shortest path. Similar to Theta*, Accelerated A* (Šišlák et al, 2009) is an any-angle path-finding algorithm that may find better path than Theta*. But it could include a large proportion of nodes in the Closed List, i.e., large number of node evaluations for challenging problems (Harabor and Grastien, 2013).

The majority of existing path-finding algorithms are for applications such as video games and navigations where computational efficiency, i.e., finding a sub-optimal solution instantly, is the most important factor. It is worth noting that the requirement on path-finding for additive-manufacturing oriented design optimization is considerably different. In the optimization of a piping network, the optimality of the path becomes critically important, while the computational efficiency takes a second role as long as the time it takes remains to be reasonable for design iteration. A known approach of finding the shortest paths in known terrain with polygonal obstacles is A* on Visibility Graphs (Lozano-Pérez and Wesley, 1979). While A* on Visibility Graphs is guaranteed to identify the shortest path, the computational cost is extremely high because it propagates along the edges of a visibility graph, the number of which grows exponentially with the size of the gridded map. Moreover, for different starting node and target node setups in the same map, visibility graphs have to be generated repeatedly. Even though the process can be accelerated by performing visibility check dynamically or using reduced

visibility graphs (Mitchell and Papadimitriou, 1991; Liu and Arimoto, 1992), A* on Visibility Graphs is still very slow and cannot be used directly in piping design as the analysis has to be carried out reiteratively due to multiple connections in each iteration of the system. Path-planning algorithms such as continuous Dijkstra and its variants (Mitchell et al, 1987; Hershberger and Suri, 1999) as well as the recent Anya (Harabor and. Grastien, 2013) also attempt to find the shortest path but have not yet been thoroughly evaluated. Generally speaking, to balance between optimality and efficiency, one would need to design an any-angle path-finding algorithm that can find a path shorter than that of the basic A* on Grids and at the same time be faster than A* on Visibility Graphs.

1.1.3 Reinforcement learning hyper-heuristic

Multi-objective optimization algorithms have been practically applied to a variety of problems, ranging from production scheduling (Wang et al, 2014; Lu et al, 2016), structural design (Kaveh and Laknejadi, 2013), performance improvement (Szollos et al, 2009), to structural fault pattern recognition (Cao et al, 2018a; 2018b) etc. The solution techniques, nevertheless, are often devised and evaluated for specific problem domains, which not only require in-depth understanding of the problem domain involved but are also difficult to be exercised to different instances. Even for the same type of problems, the formulation may need to be changed as more knowledge and insights are gained. The hyper-heuristic concept was therefore suggested (Cowling et al, 2000), aiming at producing general-purpose approaches. The terminology implies that a high-level scheme to select heuristic operators is incorporated as the detailed algorithms are being executed (Burke et al, 2009) given a particular problem and a number of low-level heuristics. Instead of finding good solutions, hyper-heuristic is more interested in adaptively finding good solution methods. Since its emergence, the subject has gained significant interests, and a number of studies of hyper-heuristic have been performed for multi-objective problems. Burke et al (2007) and Sabar et al (2011) proposed hyper-heuristic approaches to address timetabling and scheduling problems. Gomez and Terashima-Marin (2010), de Armas et al (2011) and Bai et al (2012) extended the hyper-heuristic method to handle packing and space allocating problems. Raad et al (2010) and

McClymont and Keedwell (2011) used hyper-heuristics to water resource and distribution problems. Wang and Li (2010) and Vazquez-Rodrigues and Petrovic (2013) also applied hyper-heuristic framework to multi-objective benchmark problems such as DTLZ and WFG. More recently, Guizzo et al (2015) applied hyper-heuristic based multi-objective evolutionary algorithms to solve search-based software engineering problems. Hitomi and Selva (2015; 2016) investigated the effect of credit definition and aggregation strategies on multi-objective hyper-heuristics and used it to solve satellite optimization problems. Interested readers may refer to (Burke et al, 2013; Maashi et al, 2015) for more discussions about hyper-heuristic techniques and applications.

Typically, a hyper-heuristic framework involves: (1) a high-level selection strategy to iteratively select among low-level heuristics based on the performance; (2) a predefined repository of low-level heuristics; and (3) applying the selected heuristics into optimization and evaluating their performance. The selection mechanism in hyper-heuristics, which essentially ensures the objectivity, specifies the heuristic to apply in a given point of optimization without using any domain information. With this in mind, online learning hyper-heuristics usually take advantage of the concept of reinforcement learning for selection (Kaelbling et al, 1996; Ozcan et al, 2012), as it aims to iteratively solve the heuristics selection task by weight adaptation through interactions with the search domain. The low-level heuristics correspond to a set of exploration rules, and each carries a utility value. The values are updated at each step based on the success of the chosen heuristic. An improving move is rewarded, while a worsening move is punished. The low-level heuristics can be embedded in single point search techniques, which are highly suited for these tasks because only one neighbor is analyzed for a choice decision (Nareyek, 2003). In a single point search-based hyper-heuristic framework, e.g., a simulated annealing (Kirkpatrick et al, 1983) based hyper-heuristic, an initial candidate solution goes through a set of successive stages repeatedly until termination.

1.1.4 Impedance-based structural health monitoring

Extensive research has been conducted on structural health monitoring (SHM) to protect structures from catastrophic failures. Many structural health monitoring (SHM) systems use structural dynamic responses measured by sensors to elucidate the health condition. A traditional class of methods is vibration-based SHM, which analyzes and interprets measureable modal properties such as natural frequencies and mode shapes to determine damage locations and severities (Kim and Stubbs, 2003; Maity and Tripathy, 2005; Jiang et al, 2006; Jassim et al, 2013; Cao et al, 2017). However, in practical situations, normally only lower-order modes with large wavelengths can be realistically excited and measured; thus, these methods may not be sensitive to small-sized damage (Kim and Wang, 2014). Another well-known class of methods is based on wave propagation, which uses the change of transient wave upon its passage through damage site to infer damage occurrence (Michaels and Michaels, 2007; Harley and Moura, 2014). While these high-frequency methods may entertain high detection sensitivity, it is generally difficult to use the transient responses to identify damage accurately, especially to quantify the severity of damage (Cawley and Simonetii, 2005). Piezoelectric transducers are frequently used in wave propagation-based SHM. The two-way electro-mechanical coupling of piezoelectric transducers has also allowed them to be used in piezoelectric impedance- or admittance-based methods (Park et al, 2003; Yang et al, 2008; Lim and Soh, 2012; Min et al, 2012; Annamdas and Radhika, 2013; Lim and Soh, 2014; Shuai et al, 2017). In these methods, frequency-swept harmonic voltage excitations are applied, and stationary wave responses are induced and sensed. As such, the impedance of the structure is coupled with that of the piezoelectric transducer. The change of piezoelectric impedance signature with respect to that under the undamaged baseline can be used as damage indicator.

When a credible, first-principle-based baseline model, such as the finite element model, of the healthy structure is available, we may be able to identify both the fault location and severity by using vibration responses or impedance/admittance responses which are stationary (Xia and Hao, 2003; Jiang et al, 2006; Wang and Tang, 2008; Zhou and Zuo, 2012; Shuai et al, 2017). In such an inverse analysis, we typically

divide the structure into a number of segments and assume that each segment in the model is susceptible of damage occurrence, i.e., certain property of each segment is an unknown. The inverse analysis uses the changes of stationary responses, such as natural frequencies, mode shapes, response amplitudes, impedances or admittances, as inputs. Indeed, a linearized sensitivity matrix can be derived that links the segment property change vector with respect to the response measurement change vector. This seemingly straightforward formulation appears to be appealing. Nonetheless, how to solve for the unknowns, i.e., damage locations and severities, remains a fundamental challenge. The aforementioned inverse formulation may easily become under-determined (Kim and Wang 2014). Moreover, the inevitable measurement noise and modeling uncertainty further compound the difficulty (Shuai et al, 2017).

To avoid the direct inversion of the sensitivity matrix, alternatively, the problem of identifying damage location/severity in a finite element-based analysis can be cast into a global optimization formulation. Indeed, under the umbrella of optimization formulation, several global optimization techniques, such as particle swarm optimization (Begambre and Laier, 2009), differential evolution algorithm (Seyedpoor et al., 2015), genetic algorithm (Perera et al., 2010), and DIRECT algorithm (Cao et al, 2017), have been attempted, where possible property changes in all segments are treated as unknowns to be solved. One important feature of these optimization-based formulations is that usually only forward-analysis of the model is involved to facilitate comparison and minimization of the difference between model prediction in the parametric space and actual measurements. Nevertheless, given that the sensitivity matrix is under-determined, the fundamental mathematical challenge remains, i.e., there are in theory infinitely many solutions. Inevitably, many solutions that are different from the true damage scenario may be obtained. Certainly, one would hope to remove some or even most of these 'untrue' solutions by imposing constraints in the optimization framework. In a wider context, such ill-posed problems have been tackled in mathematical and statistical literature by invoking the sparsity condition (Tarantola, 2005; Kaipio and Somersalo, 2006). For example, compressed sensing in signal processing is facilitated by applying the sparsity constraint into a usually ill-posed optimization problem (Candes et al,

2006; Donoho 2006; Mascarenas et al, 2013). Interestingly, the sparsity condition is also applicable to damage identification problem. Specifically, in practical situation, damage occurrence is much more probable to affect only a small number of elements/segments in a finite element model of the structure. In other words, the unknown vector in the damage identification problem is sparse. However, few investigations have so far taken advantage of the inherent sparse nature of damage indices to address damage identification problems. Wang and Hao (2014) formulated a pattern recognition problem for damage identification and matched the pre-defined patterns following a compressed sensing-based scheme which uses sparsity properties and l_1 regularization on the unknown damage pattern vector. This approach cannot be easily extended to cases with many damage patterns. More recently, Huang et al (2017) adopted the sparse Bayesian learning for structural damage detection. Rather than providing a single estimate, sparse Bayesian learning provides a full posterior density function, which gives a sense of confidence of the approximation.

1.1.5 Meta-model-based structural health monitoring

The advent of many new transducer materials/devices and the advancement in microelectronics have resulted in rapid progresses in this area. On the other hand, bottlenecks and unique challenges exist. Structures are continuous media, and parameters characterizing structural faults, i.e., location and severity, are continuous variables as well. Hence, structural faults have infinitely many possible patterns/profiles with typically small characteristic lengths, which are further compounded by various uncertainties. Intuitively, the dynamic response data collected by the monitoring system must be in high-frequency range (i.e., with small wavelengths) so features of small-sized faults can be captured. The key issues thus are: 1) how to effectively generate high-frequency sensing data; and 2) how to efficiently and accurately identify fault location and severity from the data (Zhang et al, 2017).

Owing to their two-way electro-mechanical coupling, piezoelectric transducers are commonly used in structural health monitoring (Wang and Tang, 2008; Gao et al, 2018), as reviewed in Section 1.1.4. However, a major hurdle remains. In theory, identifying directly the fault location and severity from

stationary responses such as impedances/admittances is possible, as long as a credible first-principle model such as finite element model of the healthy baseline is available. A linearized sensitivity matrix can be derived that links the structural property changes to the changes of harmonic response magnitudes measured. In reality, such an inverse problem is usually severely under-determined. In order to characterize high-frequency impedance/admittance responses accurately, the finite element model must have high dimensionality. To pinpoint fault condition, we often divide the structure into a number of segments where the structural property in each segment is an unknown to be solved (because each segment is susceptible of fault occurrence). Therefore, the model has high dimensionality with a large number of unknowns. Meanwhile, structural faults manifest themselves in structural resonances and anti-resonances. As such, the effective measurements of impedance/admittance changes are limited (Shuai et al, 2017). One potential way to avoid the direct inversion is to convert the identification problem into an optimization problem, where possible property changes in all segments are treated as design parameters. These parameters are updated by minimizing the discrepancy between sensor measurements and model predictions through various optimization techniques in which only forward analyses are performed (Begambre and Laier, 2009; Perera et al, 2010; Cao et al, 2018a and 2018b). The necessary computational cost, however, could be very high. The forward optimization generally requires large number of iterations to converge, while a single run of high-dimensional finite element analysis can be very costly already.

Dynamic response calibration, as a faster alternative to exhaustive finite element analysis, has shown promising aspects in alleviating computational burden by emulating the full-scale finite element model responses. Traditional response surface methods applied for model updating use explicit functions to represent the relation between inputs and outputs. Least square-based techniques are then devised to refine parameters in the polynomial representation (Ren and Chen, 2010; Fang and Perera, 2011; Li and Law, 2011; Chakrarborty and Sen, 2014). More recently, Gaussian process, also referred to as Kriging (Kennedy and O'Hagan, 2001; Rasmussen and Williams, 2006), has gained popularity due to its capability to simulate complicated process subjected to uncertainties. A Gaussian process model is not

restricted to certain polynomial form thus allows highly flexible modeling in input-output relation based on statistical expectations and variances over functions. Gao et al (2013) used a Kriging surrogate model to calibrate frequency responses for crack tip location identification in cantilever plates. Yang et al (2017) proposed a similar calibration approach in frequency domain to detect the location and severity of fault in small structures. Wan and Ren (2015) suggested a residual-based Gaussian process model to characterize the relation between residual and updated parameters in frequency domain for finite element model updating. Jin and Jung (2016) formulated a sequential surrogate modeling scheme that constructs multiple response surfaces for finite element model updating. Balafas et al (2018) presented a Gaussian process model in wavelet domain that can infer damage through hypothesis testing. It is worth noting that all these dynamic response calibration methods are applied to natural frequency measurements. Since in practical situation only lower-order natural frequencies can be realistically measured, the case setups in these studies are relatively simple with low dimensionality and the design parameters are discrete with low dimensionality as well. In comparison, in impedance/admittance sensing, considerably more amount of measurements at many frequency points, can be acquired, and a high-dimensional structure is to be identified.

From the underlying physics standpoint, impedance/admittance sensing offers a new opportunity to identify fault parameters more accurately for more complex structures. While the response calibration technique appears to be promising in possibly avoiding iterative finite element analyses in an optimization framework, new issue arises. Although fault effects are reflected in impedance/admittance change at each frequency point theoretically, the actual impedance/admittance measurements respond to a fault condition differently at different frequencies. Therefore, in order to correctly identify fault conditions, one would need to examine the impedance/admittance changes at many frequency points. In other words, in order to take full advantage of the high-frequency impedance/admittance sensing, we need to formulate and then solve efficiently an optimization problem to match response predictions with measurements at many frequency points. It should be noted that many-objective global optimization usually features more than three objectives, while multi-objective optimization refers to that with no more than three objectives.

Although it would appear to be easier to resort to weighted summation to solve a single objective optimization (Gao et al, 2013; Wan and Ren, 2015; Yang et al, 2017), weighting selection among objectives is ad-hoc, and the result could easily converge to a meaningless outcome due to multiple local optima, measurement noise and uncertainties.

1.1.6 Machine learning enabled gear fault diagnosis

The practice of fault diagnosis of gear system using vibration signals has proved to be a very challenging subject. The mainstream of gear condition monitoring is built upon various feature extraction methods that are manual and empirical in nature (Kang et al, 2001; Randall, 2011; Marquez et al, 2012). Generally, a certain signal processing technique is applied to vibration signals to identify fault-related features that are selected based on engineering judgment. Subsequently, a classifier is developed and applied to new signals to predict fault occurrence in terms of type and severity. There have been extensive and diverse attempts in manually and empirically identifying and extracting useful features from gear vibration signals, which fall into three main categories: time-domain analysis (Zhou et al, 2008; Parey and Pachori, 2012), frequency domain-analysis (Fakhfakh et al, 2005; Li et al, 2015; Wen et al, 2015) and time-frequency analysis (Tang et al, 2010; Chaari et al, 2012; Yan et al, 2014; Chen and Feng, 2017; Zhang and Tang, 2018). Time-domain statistical approaches can capture the changes in amplitude and phase modulation caused by faults (Parey and Pachori, 2012; Pachaud et al, 1997). In comparison, spectrum analysis may extract the features more easily to detect distributed faults with clear sidebands (Fakhfakh et al, 2005; Wen et al, 2015; Qian and Chen 1999). To deal with noise and at the same time utilize the transient components in vibration signals, many efforts have focused on joint time-frequency domain analysis utilizing Wigner-Ville distribution (Tang et al, 2010; Baydar and Ball, 2001), short time Fourier transform (Chaari et al, 2012; Bartelmus and Zimroz, 2009), and various wavelet transforms (Yan et al, 2014; Lin and Zuo 2003). The time-frequency distribution in such analysis can in theory lead to rich analysis results regarding the time- and frequency-related events in signals.

Although the manual and empirical methods of feature extraction have seen various levels of

successes, obviously their effectiveness is hinged upon the specific features adopted in the diagnostic analysis. It is worth emphasizing that the choices of features as well as the often-applied signal preprocessing techniques are generally based on domain expertise and subjective decisions on a specific gear system. For example, while wavelet transforms have been popular and it is well known that each wavelet coefficient can be interpreted as the energy concentration at a specific time-frequency point, it is evident from large amount of literature that there does not seem to be a consensus on what kind of wavelet to use for gear fault diagnosis. This should not come as a surprise. On one hand gear faults occur primarily at microstructure or even material level but their effects can only be observed indirectly at a system level; consequently there exists a many-to-many relationship between actual faults and the observable quantifies (i.e., features) for a given gear system (Lu et al, 2012). On the other hand, different gear systems have different designs which lead to very different dynamic characteristics. As such, the result on features manually selected and, to a large extent, the methodology employed to extract these features for one gear system design may not be easily extrapolated to a different gear system design.

Fundamentally, condition monitoring and fault diagnosis of gear systems belongs to the general field of pattern recognition. The advancements in related algorithms along with the rapid enhancement of computational power have trigged the wide spread of machine learning techniques to various applications. Most recently, deep neural network-based methods are progressively being investigated. When the parameters of a deep neural network are properly trained by available data, representative features can be extracted in a hierarchy of conceptual abstractions, which are free of human interference compared to manual selection of features. Some recent studies have adopted such type of approaches in gear fault diagnosis, aiming at identifying features implicitly and adaptively and then classifying damage/fault in an automated manner with minimal tuning. For example, Zhang et al (2015) developed a deep learning network for degradation pattern classification and demonstrated the efficacy using turbofan engine dataset. Li et al (2016) proposed a deep random forest fusion technique for gearbox fault diagnosis which achieves 97.68% classification accuracy. Weimer et al (2016) examined the usage of deep convolutional neural network for industrial inspection and demonstrated excellent defect detection results. Ince et al

(2016) developed a fast motor condition monitoring system using a 1-D convolutional neural network with a classification accuracy of 97.4%. Abdeljaber et al (2017) performed real-time damage detection using convolutional neural network and showcased satisfactory efficiency.

Deep neural network is undoubtedly a powerful tool in pattern recognition and data mining. As an end-to-end hierarchical system, it inherently blends the two essential elements in condition monitoring, feature extraction and classification, into a single adaptive learning frame. It should be noted that the amount of training data required for satisfactory results depends on many aspects of the specific problem being tackled, such as the correctness of training samples, the number of pattern classes to be classified, and the degree of separation between different classes. In most machinery diagnosis investigations, the lack of labeled training samples, i.e., experiment data of known failure patterns, is a common issue, because it is impractical to collect experimental data of each failure type and especially severity for a machinery system. To improve the performance given limited training data, some recent studies have attempted to combine preprocessing and data augmentation techniques, e.g., discrete wavelet transform (Saravanan and Ramachandran, 2010), antialiasing/decimation filter (Ince et al, 2016), and wavelet packet transform (Li et al, 2016), with neural networks for fault diagnosis. Nevertheless, the preprocessing techniques employed, which are subjected to selection based on domain expertise, may negatively impact the objective nature of neural networks and to some extent undermines the usage of such tools.

1.2 Problem statement and approach overview

By reviewing the state-of-the-art literatures, one can conclude that the performance of data-assisted analysis essentially depends on the robustness and efficiency of algorithms employed. In order to enhance the design and identification performance, this dissertation aims to address several critical issues in current research projects, which are listed following,

- To establish a new, systematic framework to tackle the type of configuration optimization problems with multiple hard constraints. Firstly, a compact mathematical model is formulated, for a representative configuration design problem with many degrees of freedom and hard

constraints. Based on the characteristics of the problem, investigations into heuristic multi-objective optimization algorithms and the treatment of hard constraints are conducted, followed by the development of an enhanced multi-objective simulated annealing algorithm called MOSA/R. The novelty of this algorithm lies in the newly designed re-seed scheme which enables the algorithm to solve the configuration optimization problem as a multi-objective optimization problem much more efficiently than existing algorithms.

- Proposed a new Focal Any-Angle A* approach. A pruned visibility graph based on Candidate Vertices, a term we use to represent the vertices that the optimal path may pass through, is developed first, followed by a ray-casting-based visibility check technique. The new approach preserves the optimality advantage of visibility graph-based methods, and reduces the number of evaluations needed for path-finding compared to both gird-based (A* on Grids, Theta*) and visibility graph-based (A* on Visibility Graphs) methods. Eventually, the developed algorithm is applied to piping network design problems where optimal paths are obtained under reasonable computational time. While the performance of some piping networks may not be merely related to the length of the pipes, we also proposed one possible approach to handle other physical or geometrical constraints. The overall design methodology has been successfully validated through a networking component prototype fabricated by Stereolithography (SLA).
- To advance the state-of-the-art in Multi-Objective Simulated Annealing (MOSA) by incorporating hyper-heuristic systematically to improve both the generality and performance. We develop a reinforcement learning hyper-heuristic inspired by probability matching (Goldberg, 1990), which consists of a selection strategy and a credit assignment strategy. Here in this research the re-seed schemes are treated as the low-level heuristics, empowering the algorithm to cover various scenarios. The main contributions are 1) to introduce a new reinforcement learning hyper-heuristic framework based on MOSA with re-seed; 2) to devise a new credit assignment strategy in high-level selection for heuristic performance evaluation; and 3) to provide insights on benchmark case studies and application.

- Develop a multi-objective optimization formulation that can be effectively solved by a deterministic global approach originated from the Dividing RECTangles (DIRECT) algorithm (Jones et al, 1993). As will be shown comprehensively later in this dissertation, a proper choice of sparse regularization under the multi-objective formulation (i.e., treating the sparsity as the second objective function) has the potential of yielding a *small* set of solutions that fits better the true damage scenario. In comparison, if a single composite objective function is used, it yields one single solution which however may not capture the true damage at all. DIRECT algorithm is particularly suitable in solving the multi-objective optimization problem formulated. Mathematically, it is a deterministic technique, and thus the results obtained are repeatable and conclusive without ambiguities. Only one algorithmic parameter, the number of function evaluations, is needed here. To enhance the computational efficiency, a new sampling/division scheme in the unknown parametric space is established.
- Develop a new methodology of fault identification using piezoelectric impedance/admittance sensing. To thoroughly elucidate the health status, a many-objective optimization is formulated to match parametric prediction with measurements at all frequency points of interest. Gaussian process regression is incorporated to construct the response surfaces, which not only significantly reduces computational cost but also yields continuous searching of fault parameters. Our goal in optimization is to find many solutions (owing to the under-determined nature of the problem) that are all optimal. In order to balance between solution convergence and diversity, we establish an ε-dominance enabled many-objective simulated annealing algorithm. Subsequently, inspired by concepts in social statistics, i.e., voting power and majority voting (Taylor and Pacelli, 2008), a voting score calculation framework is employed to evaluate quality of the solutions obtained. As a combination of many-objective optimization and voting score calculation, our proposed many-objective evaluation approach is able to distinguish the solutions that could accurately indicate the health condition of the structure and ultimately provide guidance for further examination.
- Present a deep neural network-based transfer learning approach utilizing limited time-domain

data for gearbox fault diagnosis. One-dimensional time-domain data of vibration responses related to gear fault patterns are converted into graphical images as input. The approach inherits the non-biased nature of neural networks that can avoid the manual selection of features. Meanwhile, the issue of limited data is overcome by formulating a new neural network architecture that consists of two parts. Massive image data (1.2 million) from ImageNet (http://www.image-net.org/challenges/LSVRC/2010/) are used first to train an original deep neural network model, denoted as neural network A. The parameters of neural network A are transferred (copied) to the new architecture as the first part. The second part of the architecture, an untrained neural network B, accommodates the gear fault diagnosis task and is further trained using experimentally generated gear fault data. Unlike traditional neural networks, the training set of transfer learning do not necessarily subordinate to the same category or from the same physical background (Yang et al, 2018). As to be demonstrated later, with this new architecture, highly accurate gear fault diagnosis can be achieved using limited time-domain data directly without involving any subjective preprocessing techniques to assist feature extraction.

Chapter 2. Harnessing Multi-Objective Simulated Annealing toward Configuration Optimization within Compact Space for Additive Manufacturing

The rapid advancement of additive manufacturing technology has led to new opportunities and challenges. One potential advantage of additive manufacturing is the possibility of producing systems with reduced volumes/weights. This research concerns a type of configuration optimization problems, where the envelope volume in space occupied by a number of components is to be minimized along with other objectives. Since in practical applications the objectives and constraints are usually complex, the formulation of computationally tractable optimization becomes difficult. Moreover, unlike conventional multi-objective problems, these configuration optimization problems usually come with a number of demanding constraints that are hard to satisfy, which results in the critical challenge of balancing solution feasibility with optimality. In this research, the mathematical formulation of a representative problem of configuration optimization with multiple hard constraints is first presented, followed by two newly developed versions of an enhanced multi-objective simulated annealing approach, referred to as MOSA/R, to solve this problem. To facilitate the optimization computationally, in MOSA/R, a versatile re-seed scheme that allows biased search while avoiding pre-mature convergence is designed. Our case studies indicate that the new algorithm yields significantly improved performance towards both constrained benchmark tests and constrained configuration optimization problem. The methodology developed can lead to an integrated framework of design and additive manufacturing.

2.1 Introduction

Configuration design and optimization have been studied since the Kepler Conjecture (i.e., no arrangement of equally sized spheres filling space has a greater average density than that of the cubic close packing and hexagonal close packing arrangements). In modern computer-integrated manufacturing, configuration optimizations are frequently encountered in aerospace and automotive systems (Grignon and Fadel, 2004), manufacturing facilities and plants (Yang and Hung, 2007; Gonzalez-Cruz and

Martinez, 2011; Nageshwaraniyer et al, 2013), and 3-dimensonal laser cutting (Gan et al, 2004) etc. In recent years, new opportunities and new challenges unfold with the emergence of additive manufacturing (Lipson and Kurman, 2013; Cao et al, 2018). Additive manufacturing is capable of producing systems with reduced volumes/weights. Traditionally, configuration optimization has been carried out manually based on empirical knowledge due to a large number of constraints associated with manufacturing access requirements, which is time-consuming and may hinder the result from being optimal or even sub-optimal. However, with the higher design freedom brought by additive manufacturing, the components now can be manufactured as designed, which offers a new time-efficient convenience to truly facilitate a streamlined process of design automation and manufacturing optimization. Our work is derived from such opportunity and motivated by the engineering problem of how to spatially arrange a number of components (e.g., cylinders) within a highly restricted space in order to minimize the envelope volume of the assembly as well as the distance between these components that have connectivity relations while satisfying certain prescribed constraints.

In general, configuration design involves a wide variety of goals and objectives, and densest packing is a good example in terms of the difficulties one may encounter when dealing with such topics. In two-dimensional scenarios, one is given a set of geometries such as rectangles, polyminos or spheres. The goal is to pack these items orthogonally into a single rectangular box of unlimited height which needs to be minimized (Bortfeldt and Wäscher, 2013), or alternatively, to pack a number of circles inside a circumcircle whose radius needs to be minimized (Muller et al, 2009). Three-dimensional problems can be defined in a similar fashion, e.g., given a set of three-dimensional objects of arbitrary geometry and an available space (possibly the space of a container), find a placement for the objects within the space that achieves the design objectives, such that none of the objects interferes (i.e. occupy the same space) while optional spatial and performance constraints on the objects are satisfied. The problem of packing has shown to be NP-complete (Garey and Johnson, 1979), i.e., no optimal algorithm is known running in polynomial time. Therefore, even simple design problems involving spheres, squares or rectangles are

known to be difficult problems in the mathematical sense (Grignon and Fadel, 2004). As such, heuristic approaches are required to obtain near-optimal solutions in realistic amount of computational time.

Throughout the years, various global optimization strategies have been tailored to address the configuration optimization problems. Dyckhoff (1990) used branch and bound approach to solve simple rectangular layout problems with a small number of objects. Szykman and Cagan (1995) presented a Simulated Annealing (SA) based approach to generate configurations. Although theoretically general, their work was limited to simple geometric shapes and restricted rotations. Then the work was extended to configuration problems of arbitrary shapes by considering the spatial constrains between components using the same SA approach (Cagan et al, 1998). Meanwhile, Sachdev et al (1998) proposed a framework to integrate techniques such as Simulated Annealing and Genetic Algorithms (GA) to solve configuration design problems. Grignon and Fadel (2004) applied multi-objective GA towards configuration optimization problems with more complex objectives. Later on, Aiello et al (2006) employed a multi-objective constrained genetic algorithm to solve facility layout problems. A recent review of packing and configuration design methods can be found in (Fadel and Wiecek, 2015). It is worth noting that in previous investigations, the feasibility of the design, i.e., finding a configuration that satisfies constraints, has not been a major challenge. On the other hand, many configuration design problems in modern engineering practices, such as the manifold configuration design problem, are subjected to multiple hard constraints derived from practical requirements, e.g., the distance between certain components is restricted, and neither overlap nor out-of-bound is allowed. In these scenarios, finding a configuration that satisfies these constraints itself may become computationally demanding. Intuitively, constraints satisfaction should be added to the priority list in optimization along with convergence and diversity.

The form of configuration optimization problems varies for specific applications. For example, as will be shown later, the three-dimensional problem of interest is abstracted from certain manifold configuration design problems, is a cylinder-based problem with multiple objectives and constrains. Nevertheless, different configuration optimization problems could be tackled, in a similar manner, as global optimization problems which are usually nonlinear and multi-modal. In this research, our goal is

to establish a new, systematic framework to tackle the type of configuration optimization problems with multiple hard constraints. Firstly, a compact mathematical model is formulated, for a representative configuration design problem with many degrees of freedom and hard constraints. Based on the characteristics of the problem, investigations into heuristic multi-objective optimization algorithms and the treatment of hard constraints are conducted, followed by the development of an enhanced multi-objective simulated annealing algorithm called MOSA/R. The novelty of this algorithm lies in the newly designed re-seed scheme which enables the algorithm to solve the configuration optimization problem as a multi-objective optimization problem much more efficiently than existing algorithms. The rest of the chapter is organized as follows. In Section 2.2, a comprehensive mathematical model of the problem is formulated, where the unique features of such optimization problem are highlighted. It is recognized as a constrained non-differentiable multi-objective optimization (MOO) problem. Subsequently, Section 2.3 gives an overview of representative techniques to solve constrained MOO problems, followed by discussions on their respective pros and cons. A new algorithm, referred to as Multi-Objective Simulated Annealing based on Re-seed (MOSA/R), is then developed in Section 2.4 to specifically solve the problem. Implementation details and numerical tests results are presented in Sections 2.5 and 2.6 respectively, where performance metrics are defined and the performance of two versions of MOSA/R is evaluated and compared to some well-known multi-objective algorithms (MOEA/D, NSGA-II and AMOSA) when applied to benchmark tests and the configuration optimization problem.

2.2 Problem definition

2.2.1 Case setup

Figure 2.1 illustrates a recent manifold configuration design problem we have facilitated using the approach proposed in this study. To enhance the reproducibility of our study, we abstract the actual design problem into a representative mathematical formulation.

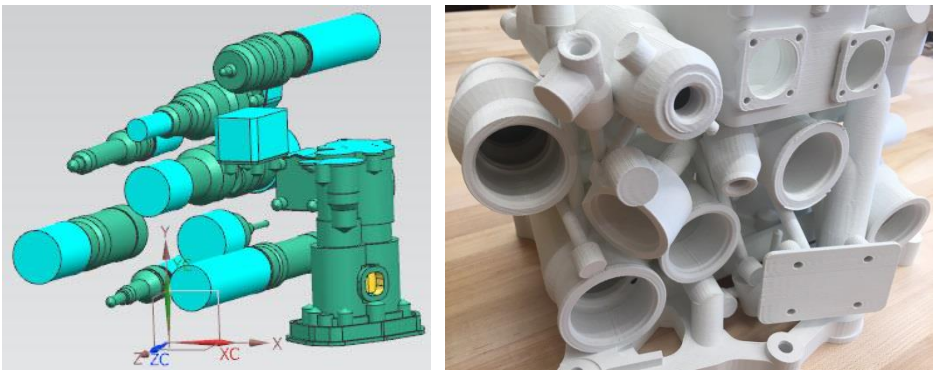


Figure 2.1 Automated manifold configuration design (a) Components to be placed within a restricted manifold (b) Configuration prototype produced using additive manufacturing.

The abstracted model involves six cylinders, which are viewed as six functional units that need to be placed within restricted space, with full rotational and translational degrees of freedom. Free-form object representation is precluded to avoid inefficient interference computation. Accordingly, all the cylinders are confined within a cubic space that has a side length of *SL* in a Cartesian system and each side of the cubic space parallels to one of the axes. The surface of the envelope is defined by the largest coordinate value that the cylinder bodies can reach. Note that the surfaces of the cubic space are not the surfaces of the assembly envelope. A representative cylinder is illustrated in Figure 2.2. One example configuration is depicted in Figure 2.3. The dimension of each cylinder is defined by its diameter 2*r* and length (Table 1). The connectivity relations of these cylinders are provided in Table 2.2. The design objectives of this test case are: optimizing the volume of the envelope, and meanwhile optimizing the distance of windows that have a connective relationship.

In addition to the aforementioned design objectives, the following constraints are also given:

Constraint 1: Cylinder 1 base must contact cube surface ‘a’;

Constraint 2: Cylinder 6 base must contact cube surface ‘b’;

Constraint 3: Envelope must be located inside the cube, i.e., no cylinder is allowed to surpass the cuboid surfaces;

Constraint 4: Connective line 3 must be 5 inches maximum;

Constraint 5: Connective line 4 must be 3 inches maximum;

Constraint 6: There must be at least 0.5 inch spacing between cylinders.

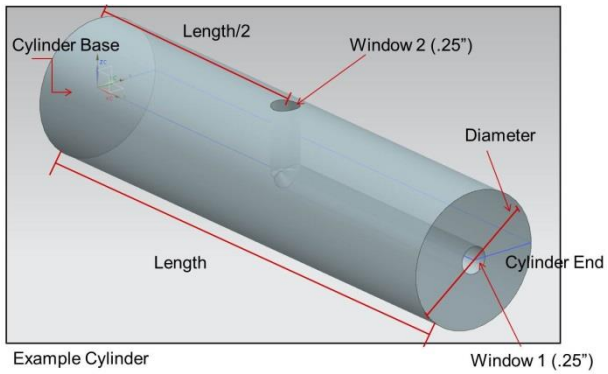


Figure 2.2 An example cylinder.

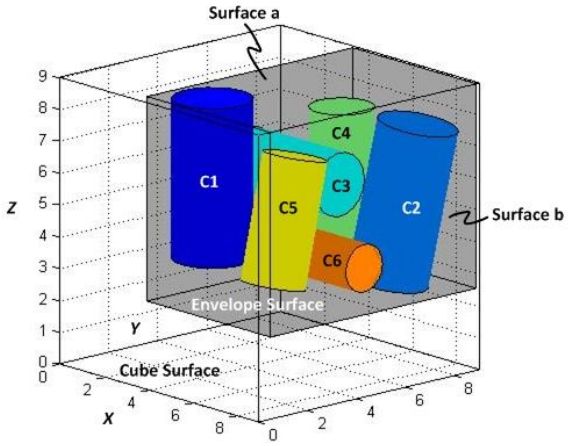


Figure 2.3 An example configuration.

The six-cylinder model used in this paper is extracted from an engineering design problem. As each cylinder is allowed to move and rotate, the optimization problem has large design space. Moreover, a critical challenge here is that within such a large design space, the feasible region of design is actually small, because the constraints are demanding and would be easily violated. This test problem can represent a type of configuration optimization problems featuring large design space and constraints that are hard to satisfy. In the numerical experiments, the boundary constraints (i.e. *SL*) are varied so that different level of constraints is considered.

Table 2.1 Specifications of cylinders

Component	Diameter 2*r* (inch)	Length *l* (inch)
Cylinder 1	1.25	5

Cylinder 2	1.25	5
Cylinder 3	1.00	4
Cylinder 4	1.00	4
Cylinder 5	1.00	4
Cylinder 6	0.75	3

Table 2.2 Connectivity relation between cylinders

Connective Line	Start Point	End Point
1	Window 2 of Cylinder 1	Window 1 of Cylinder 6
2	Window 2 of Cylinder 6	Window 1 of Cylinder 2
3	Window 2 of Cylinder 2	Window 1 of Cylinder 4
4	Window 2 of Cylinder 4	Window 1 of Cylinder 3
5	Window 2 of Cylinder 3	Window 1 of Cylinder 5
6	Window 2 of Cylinder 5	Window 1 of Cylinder 1

2.2.2 Mathematical formulation of the optimization problem

As the diameter and length of each cylinder have been specified, five decision variables x, y, z, θ and φ are used to designate the exact position and direction of each cylinder, where (x, y, z) represents the absolute location of the center of cylinder base in the Cartesian coordinate system, and (θ, φ) represents the direction of the cylinder using spherical coordinate notations (Figure 2.4). With pre-specified parameters r and l, a cylinder can be fully described by (x, y, z, θ, φ, r, l). Accordingly, the coordinate of the center of cylinder end can be expressed as

$$\bar{x} = x + l\sin\theta\cos\phi, \qquad \bar{y} = y + l\sin\theta\sin\phi, \qquad \bar{z} = z + l\cos\theta \tag{2.1a-c}$$

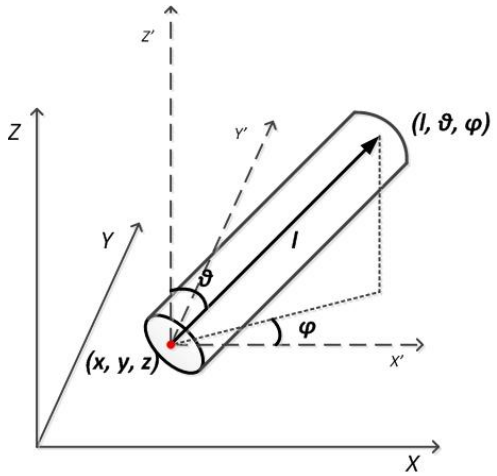


Figure 2.4 Cylinder in coordinate system.

We assume X_e, Y_e, and Z_e are the lengths of envelope sides that parallel with the *X*- , *Y*-, and *Z*-axis, and *d(i, j)* is the Euclidean distance between the midpoint of axis of Cylinder *i* and Window 1 of Cylinder *j* (i.e., the approximated length of connective line between Window 2 of Cylinder *i* and Window 1 of cylinder *j*). The term 'window' here refers to a specific location on the cylinder that can serve as either the starting point or the destination of a connective line (Figure 2.2). The *optimization objectives* are

$$\text{Min (Envelope Volume)} = X_e \cdot Y_e \cdot Z_e \tag{2.2}$$

$$\text{Min (Connective Line Length)} = \sum_{id=1}^{6} d(i, j)_{id} \tag{2.3}$$

If the maxima and the minima that the body of Cylinder *k* can reach in the *X*-, *Y*-, and *Z*-axis directions are $x_k^{\max}$, $x_k^{\min}$, $y_k^{\max}$, $y_k^{\min}$, $z_k^{\max}$ and $z_k^{\min}$ respectively. We have

$$\begin{cases} x_k^{\max} = x_k + r_k \sin\theta_k \cos\phi_k \\ x_k^{\min} = \bar{x}_k - r_k \sin\theta_k \cos\phi_k \end{cases} \quad \text{when } x_k \geq \bar{x}_k$$

$$\begin{cases} x_k^{\max} = \bar{x}_k + r_k \sin\theta_k \cos\phi_k \\ x_k^{\min} = x_k - r_k \sin\theta_k \cos\phi_k \end{cases} \quad \text{when } x_k < \bar{x}_k$$

$$\begin{cases} y_k^{\max} = y_k + r_k \sin\theta_k \sin\phi_k \\ y_k^{\min} = \bar{y}_k - r_k \sin\theta_k \sin\phi_k \end{cases} \quad \text{when } y_k \geq \bar{y}_k$$

$$\begin{cases} y_k^{\max} = \bar{y}_k + r_k \sin\theta_k \sin\phi_k \\ y_k^{\min} = y_k - r_k \sin\theta_k \sin\phi_k \end{cases} \quad \text{when } y_k < \bar{y}_k$$

$$\begin{cases} z_k^{\max} = z_k + r_k \cos\theta_k \\ z_k^{\min} = \bar{z}_k - r_k \sin\theta_k \end{cases} \quad \text{when } z_k \geq \bar{z}_k$$

$$\begin{cases} z_k^{\max} = \bar{z}_k + r_k \cos\theta_k \\ z_k^{\min} = z_k - r_k \sin\theta_k \end{cases} \quad \text{when } z_k < \bar{z}_k \tag{2.4a-f}$$

Take Equation 2.4(a) for example. Its physical meaning is that the maxima/minima for Cylinder *k* in the *X*- axis direction is specified by the *x* value of the base/end of the cylinder, the radius, as well as the

yaw angle and pitch angle (Figure 4) when $x_k \geq \bar{x}_k$. Thus, X_e, Y_e, and Z_e can be expressed in the following form,

$$X_e = \max_{i=1,\cdots,6}(x_i^{\max}) - \min_{i=1,\cdots,6}(x_i^{\min})$$

$$Y_e = \max_{i=1,\cdots,6}(y_i^{\max}) - \min_{i=1,\cdots,6}(y_i^{\min})$$

$$Z_e = \max_{i=1,\cdots,6}(z_i^{\max}) - \min_{i=1,\cdots,6}(z_i^{\min}) \qquad (2.5a\text{-}c)$$

In Equation (2.3), *id* stands for the index number of connective lines, e.g., $(i, j)_1 = (1,6)$, $(i, j)_2 = (6,2)$, as indicated in Table 2.2. $d(i, j)$ is given as

$$d(i,j) = \sqrt{[(\frac{x_i + \bar{x}_i}{2}) - x_j]^2 + [(\frac{y_i + \bar{y}_i}{2}) - y_j]^2 + [(\frac{z_i + \bar{z}_i}{2}) - z_j]^2} \qquad (2.6)$$

Next we consider the mathematical expressions of the constraints. The maxima and minima of the cube in the *X*-, *Y*-, and *Z*-axis directions are represented by $Cx_{\max}$, $Cx_{\min}$, $Cy_{\max}$, $Cy_{\min}$, $Cz_{\max}$ and $Cz_{\min}$, respectively, which are determined by the placement of the cube in the Cartesian coordinate system and *SL*. As *SL* has been defined to be the length of each side of the cube, the following must be satisfied,

$$SL = Cx_{\max} - Cx_{\min} = Cy_{\max} - Cy_{\min} = Cz_{\max} - Cz_{\min} \qquad (2.7)$$

Constraints 1 and 2 described in Section 2.1 can now be mathematically expressed as

$$z_1 \equiv Cz_{\max},$$

$$x_6 \equiv Cx_{\max},$$

$$\theta_1 = 180^o, \qquad \phi_1 = 0^o$$

$$\theta_6 = -90^o, \qquad \phi_6 = 180^o \qquad (2.8a\text{-}d)$$

which essentially indicate that the maximum values in the *Z*- and *X*- axis directions that the cylinders can reach are defined by the *z* value of Cylinder 1 and *x* value of Cylinder 6 because their bases will be in contact with cube surface 'a' and 'b'. For Constraint 3, the entire cylinder cannot pass through the cuboid surface, yields

$$\max(x_i^{\max}) \le Cx_{\max} = x_6, \quad \min(x_i^{\min}) \ge Cx_{\min}, \quad i = 1,\cdots,6$$

$$\max(y_i^{\max}) \le Cy_{\max}, \quad \min(y_i^{\min}) \ge Cy_{\min}, \quad i = 1,\cdots,6$$

$$\max(z_i^{\max}) \le Cz_{\max} = z_1, \quad \min(z_i^{\min}) \ge Cz_{\min}, \quad i = 1,\cdots,6 \qquad (2.9a\text{-}c)$$

Constraints 4 and 5 concern the connective line length

$$d(2,4) \le 5 - a$$

$$d(4,3) \le 3 - a \qquad (2.10a, b)$$

Since the connective lines are not always the shortest straight lines between windows, in Equations (2.11a) and (2.11b), *a* is a parameter that translates the connective length to Euclidean distance, which is assumed to be 1 here. *D(i, j)* is defined to be the shortest distance between the center axes of Cylinders *i* and *j*. As Constraint 6 specifies that there must be at least 0.5 inch spacing between cylinders, we have,

$$D(i,j) \ge r_i + r_j + 0.5, \; i, j = 1,\cdots,6; \; i \ne j \qquad (2.11)$$

In summary, the complete optimization problem formulation can be summarized as

Minimize envelop volume and connective line length (Equations (2.2) and (2.3))

subject to 1) contact constrains (Equation (2.8), equalities)

2) boundary constraints (Equation (2.9), inequalities)

3) connective line constraints (Equation (2.10), inequalities)

4) overlap constraints (Equation (2.11), inequalities)

This is inherently a multi-objective optimization problem, as we need to tackle two separate minimization objectives. In order to incorporate the above-mentioned inequality constraints into the analysis, penalty functions are introduced such that these constraints can be treated equivalently as additional objective functions. For Constraints 1 and 2 represented by Equations (2.9a)-(2.9c), one penalty function is defined as

$$\begin{aligned} P_1 = {} & [\max(x_i^{\max}) - Cx_{\max}]\Theta[\max(x_i^{\max}) - Cx_{\max}] + [Cx_{\min} - \min(x_i^{\min})]\Theta[Cx_{\min} - \min(x_i^{\min})] \\ & + [\max(y_i^{\max}) - Cy_{\max}]\Theta[\max(y_i^{\max}) - Cy_{\max}] + [Cy_{\min} - \min(y_i^{\min})]\Theta[Cy_{\min} - \min(y_i^{\min})] \\ & + [\max(z_i^{\max}) - Cz_{\max}]\Theta[\max(z_i^{\max}) - Cz_{\max}] + [Cz_{\min} - \min(z_i^{\min})]\Theta[Cz_{\min} - \min(z_i^{\min})] \end{aligned} \qquad (2.12)$$

where $\Theta(a) = \begin{cases} 1 & \text{if a} > 0 \\ 0 & \text{otherwise} \end{cases}$ is the Heaviside function. Mathematically, minimizing P_1 will move the corresponding result toward satisfying the original constraints till $P_1 = 0$ which means the constraints are fully satisfied. Similarly, for the remaining constraints expressed as Equations (2.10) and (2.11), similar penalty functions are introduced

$$P_2 = [d(2,4) - 5 + a]\Theta[d(2,4) - 5 + a] + [d(4,3) - 3 + a]\Theta[d(4,3) - 3 + a] \tag{2.13}$$

$$P_3 = \sum_{i=1}^{6} \sum_{j=1, j\neq i}^{6} [r_i + r_j + 0.5 - D(i,j)]\Theta[r_i + r_j + 0.5 - D(i,j)] \tag{2.14}$$

Each kind of inequality constraint has thus been translated into one objective function. It is worth noting that the equality constraints are considered in the move routine of the algorithm to be developed. Our goal now becomes to minimize 5 objective functions expressed as Equations (2.2), (2.3), (2.12), (2.13), and (2.14). We have a multi-modal optimization problem associated with 24 design variables and 5 discontinuous objectives within a very large design space.

2.3 Related studies

In this section, existing techniques that could possibly be used to solve the problem formulated in the preceding section are investigated. The problem defined involves simultaneous optimization of several incommensurable and potentially conflicting objectives. Intuitively, multi-objective optimization (MOO) could be facilitated by forming an alternative problem with a single, composite objective function using a weighted sum approach. Single objective optimization techniques are then applied to this composite function to obtain a single optimal solution. However, the weighted sum methods have difficulties in selecting proper weighting factors especially when there is no articulated *a prior* preference among objectives. Indeed, *a posteriori* preference articulation is usually preferred, because it allows a greater degree of separation between the algorithm and the decision-making process which also enables the testing process to be conducted independently of the application (Giagkiozis et al, 2015). Furthermore, instead of a single optimum produced by weighted sum methods, MOO will yield a set of alternative solutions explicitly exhibiting the tradeoff between different objectives. In light of this, significant

amount of research has been carried out on solving MOO problems. Here we start from outlining representative categories of methods of MOO.

2.3.1 Overview of MOO methods

In this sub-section we briefly overview some representative MOO methods, which provides the basis for the subsequent discussion of the handling of constraints and the development of the new algorithm.

The most well-known MOO methods are probably the Pareto-based methods that define optimality in a wider sense that no other solutions in the search space are superior to Pareto optimal solutions when all objectives are considered (Zitzler, 1999). A general Pareto-based MOO problem where *n* objectives are minimized simultaneously is

$$\text{Minimize } \boldsymbol{y} = \boldsymbol{f}(\mathbf{x}) = (f_1(\mathbf{x}),...,f_n(\mathbf{x})) \tag{2.15}$$

where $\mathbf{x} = (x_1, x_2,..., x_k) \in \mathbf{X}$ and $\mathbf{y} = (y_1, y_2,..., y_n) \in \mathbf{Y}$. $\mathbf{x}$ is the decision vector of *k* decision variables, and $\mathbf{y}$ is the objective vector. $\mathbf{X}$ denotes the decision space while $\mathbf{Y}$ is called the objective space. When two sets of decision vectors are compared, the concept of dominance is used. Assuming **a** and **b** are decision vectors, the concept of Pareto optimality can be defined as follows: **a** is said to dominate **b** if:

$$\forall i = \{1,2,...,n\}: f_i(\mathbf{a}) \le f_i(\mathbf{b}) \tag{2.16a}$$

and

$$\exists\, j = \{1,2,...,n\}: f_j(\mathbf{a}) < f_j(\mathbf{b}) \tag{2.16b}$$

Refer to Figure 2.5. Any objective function vector which is neither dominated by any other objective function vector of a set of Pareto-optimal solutions nor dominating any of them is called non-dominated with respect to that Pareto-optimal set. The solution that corresponds to the objective function vector is a member of Pareto-optimal set. Usually $\prec$ is used to denote domination relationship between two decision vectors (Table 2.3).

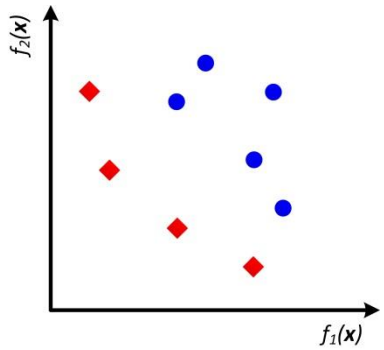


Figure 2.5 Domination relations within a set of solutions for a two objectives minimization problem (◆: Non-dominated solutions; ●: Dominated solutions).

Table 2.3 Domination relations

Relation	Symbol	Interpretation in objective space
a dominates **b**	$\mathbf{a} \prec \mathbf{b}$	**a** is not worse than **b** in all objectives and better in at least one objective
b dominates **a**	$\mathbf{b} \prec \mathbf{a}$	**b** is not worse than **a** in all objectives and better in at least one objective
Non- dominant to each other	$\mathbf{b} \cong \mathbf{a}$	**a** is worse than **b** in some objectives but better in some other objectives

Another class of methods, the decomposition-based methods, is similar to the weighted sum approach. Unlike traditional weighted sum, they employ a set of weighting vectors to decompose an MOO into a set of single-objective sub-problems. Such methods are compatible with population based techniques such as Genetic Algorithm. A well-known example was conducted by Zhang et al (2007). The third class of methods is indicator-based. They are based on metrics measuring the quality and fitness of the solution. One of the most popular indicators is HyperVolume which was introduced by Zitzler and Thiele (1999). But the calculation of HyperVolume is NP-hard itself, and the computational cost may be prohibitive when the number of objectives is large. Lastly, if *a priori* knowledge is available, which is usually not the case for practical applications, goal-attainment and ε -constraint could also be considered.

2.3.2 Constraint handling

It is worth emphasizing again that the test problem formulated in Section 2.2 comes with hard constraints. Here 'hard' means the constraints are not only must-satisfy ones but also difficult to be satisfied (which will be illustrated in Section 6 through actual simulations). Handling constraints within MOO is an important task that deserves attention particularly when dealing with practical problems that

have constraints that should be incorporated into the optimizer (Coello et al, 2007). Hence, a biased decision-making technique should be involved favoring the constraints. There are several methods of handling constraints in multi-objective optimization. One technique is to ignore infeasible solutions. This technique tries to ignore infeasible solutions along the optimization process (Cao et al, 2016), so that only a newly generated feasible solution will be taken into consideration. But in most practical applications, finding a feasible solution is a major problem itself. Another technique is to use penalty functions, where larger penalty parameters could be assigned to objectives with higher priority. A general formulation takes the form of $\phi(\mathbf{x}) = f(\mathbf{x}) \pm \left[\sum_{i=1}^{n} h_i \times G_i \right]$ where $\phi(\mathbf{x})$ is the new objective function to be optimized, G_i is the function of the constraint $g_i(\mathbf{x})$, and h_i is a positive penalty parameter. The most common form for G_i is $G_i = \max(0, g_i(\mathbf{x}))$. This method relies heavily on the proper selection of the penalty parameters. If inappropriate parameters are chosen, either a set of infeasible solutions or a poor distribution of solutions is likely. Yet it is still not clear how to select parameters scientifically to guide the search towards the most desirable direction. A third type technique can be regarded as systematic constraint handling procedure (Deb, 2000) where the solutions are carefully classified into feasible ones and infeasible ones. Similar to the challenge encounter by the first technique mentioned above, when most solutions are infeasible along the optimization process, this method would have difficulties in finding solutions.

Because the feasible region is significantly smaller as opposed to the infeasible region for problems with hard constraints, the first technique (ignoring infeasible solution) and the third technique (systematic constraint handling) seem inadequate. In addition, due to the difficulty to determine penalty parameters, the second technique (penalty function) appears to be ad-hoc. Alternatively, a potentially promising approach is the combined non-domination check suggested by Ray et al (2001). A general constrained multi-objective optimization problem (in the minimization sense) can be expressed as:

$$\text{Minimize } \mathbf{y} = \boldsymbol{f}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_n(\mathbf{x})) \tag{2.17}$$

$$\text{subject to } \mathbf{c} \equiv g_i(\mathbf{x}) \geq a_i \quad i = 1, 2, ..., q \tag{2.18}$$

Then a combined objective can be defined,

$$\mathbf{y}_{\text{combined}} = (f_1(\mathbf{x}), ..., f_n(\mathbf{x}), c_1, ...c_q) \tag{2.19}$$

Then, the non-domination check of the combined objective could be made using Equations (2.16a) and (2.16b).

The main idea behind is intuitive, e.g., a constrained single-objective optimization problem will be transformed into an unconstrained MOO problem based on Pareto optimality such that no aggregation of objectives or constraints is involved. This method eliminates the need for fine-tuning penalty parameters so one could approach the feasible region in a more efficient way (Mezura-Montes and Coello, 2006). For more details of constraint handling, one may refer to (Coello et al, 2007; Huband et al, 2006).

2.4 An improved simulated annealing algorithm: MOSA/R

The formulation of an improved algorithm to efficiently solve the configuration optimization problem is presented in this section. Figure 2.6 shows the compatibility relations between constraint handling techniques and multi-objective handling techniques. We hope to use combined non-domination check to handle constraints; the only cohesive approach is the Pareto-based approach. Thus, we decide to employ such combination. Essentially, a combined objective as demonstrated in Equation (2.19) is used as the design objective followed by the Pareto domination check as illustrated in Equations (2.16). Therefore, neither weighted aggregation nor feasibility check is involved. Genetic Algorithms (GA) and Simulated Annealing (SA) are considered as the effective approaches of such kind in solving configuration design problems (Fadel and Wiecek, 2015). In this application, SA is chosen over GA as the result of our preliminary investigation, for several reasons. First, the decision space of our test problem is larger compared to test cases like ZDT, DTLZ or WFG in literature (Huband et al, 2006). The synergy effect of large decision space and multi-modality may render two sets of approximated optimal solutions sitting very far away from each other in decision space. As suggested by Beyer (1997), the crossover operator has the effect of genetic repair which survives the common features of both parents. The crossover

operation may not be as effective towards the test problem because the parents would probably have nothing in common. The second reason is that most solutions found along the course of optimization are infeasible. GA, as a population-based algorithm, generates more infeasible solutions than that of SA when executed under the same number of evaluations, which means GA is more computationally intensive (or converges slower) than SA under such circumstance. As reported by Mann and Smith (1996), the execution time of GA is from 10 to 24 times longer than that of SA. On the other hand, SA generates one solution at a time, which is easier to manipulate towards feasibility. In fact, SA can, in a certain way, be viewed as a special case of GA (Pinedo, 2005). As will be shown later, our simulation results match the above statement well (see Section 2.6.1 and Section 2.6.2). For the configuration optimization problem analyzed in this research, the GAs (MOEA/D and NSGA-II) applied cannot find feasible solution in a reasonable time frame. Thus, the corresponding results are not included in Section 2.6.3.

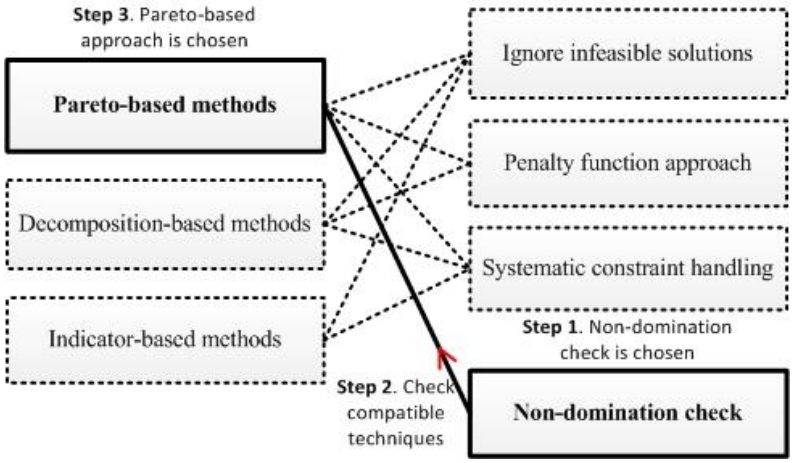


Figure 2.6 Compatibility relationships between multi-objective handling and constraint handling techniques.

Simulated Annealing (Kirkpatrick et al, 1983) is a heuristic technique drawing an analogy from physics annealing process. It was originally designed for solving single objective optimization problem and was proven to be robust and convergent if annealed sufficiently slow (Mitra et al, 1986). Engrand, who is among the very first to embed the concept of Pareto optimality with SA, proposed to maintain an external population archiving all non-dominated solutions found so far (Engrand, 1998). Then several

Multi-Objective Simulated Annealing (MOSA) methods that incorporate Pareto set (Czyzak and Jaszkiewicz, 1998; Ulungu et al, 1999; Suppapitnarm et al, 2000; Nam and Park, 2000) have been developed. The acceptance criteria of these methods are all derived from the differential between the new and current solutions. However, in the presence of Pareto set, solely comparing the new solution to the current solution appears to be vague. That is why there have been a few techniques proposed that use Pareto domination based acceptance criterion in MOSA (Suman, 2005; Smith, 2006; Bandyopadhyay et al, 2008; Zaretalab et al, 2015), the salient feature of which is that the domination status of the point is considered not only with respect to the current solution but also the archive of non-dominated solutions found so far. It has been widely demonstrated that simulated annealing algorithms are capable of finding multiple Pareto-optimal solutions in a single run.

2.4.1 Algorithm formulation

Based on the discussion above, a re-seed based MOSA algorithm is developed, hereafter referred to as MOSA/R, following the lead of AMOSA (Bandyopadhyay et al, 2008) that uses dominance measure to compute the probability of acceptance of a new point. This new MOSA/R scheme will enable constraints handling without the need to specify parameters through the use of a re-seed scheme combined with non-dominated rank of the constraints. In this research, two different versions of MOSA/R are tested. The feasibility and efficiency of the proposed methods with application to the test problem defined earlier are illustrated and compared with AMOSA. The flowchart and pseudo-code of MOSA/R is provided below.

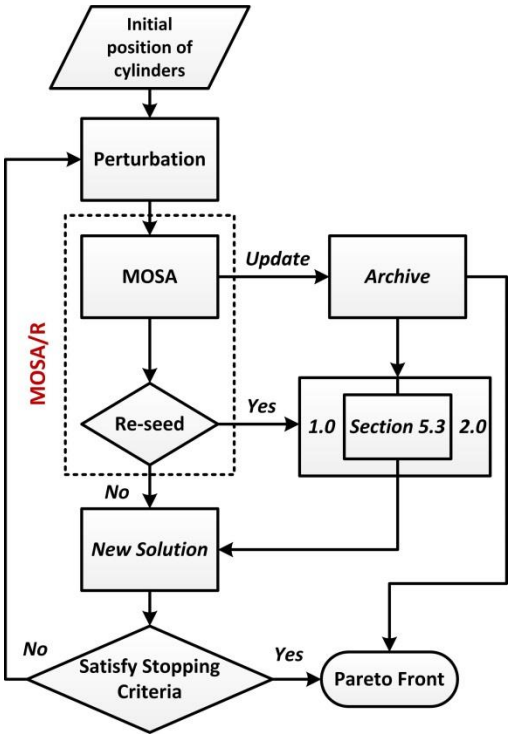


Figure 2.7 Flowchart of MOSA/R.

Algorithm MOSA/R

1. Set *Tmax*, *Tmin*, # of iterations *iter*, cooling rate α
2. Initialize the *Archive* (Pareto front)
3. *Current solution* = randomly chosen from *Archive*
4. While (*T* > *Tmin*)
5. For 1 : *iter*
6. Generate a *new solution* vector in the neighborhood of *current solution* vector
7. If *new solution* dominates *k* (*k* >= 1) solutions in the *Archive* /*Case 1*/
8. Remove all *k* dominated points from the *Archive*
9. Add *new solution* to the *Archive*
10. Set *new solution* as *current solution*
11. Else if *new solution* dominated by *k* (*k* >= 1) solutions in the *Archive* /*Case 2*/
12. If *new solution* dominated by *current solution* /*Case 2a*/
13. If *current solution* $\in$ *Archive* /*Case 2a-1*/
14. $$prob = \frac{1}{1+\exp(\Delta dom_{avg} / T)}$$
$$\text{where } \Delta dom_{avg} = \frac{\sum_{i=1}^{k} \Delta dom_{i,new}}{k}$$
15. Set *new solution* as *current solution* with *prob*
16. Else if *current solution* $\notin$ *Archive* /*Case 2a-2*/
17. Select a solution from *Archive* following designed rules
18. $$prob = \frac{1}{1+\exp(-\Delta dom_{select,new})}$$

where $\Delta dom_{select,new}$ = domination amount between *selected solution* and *new solution*

19. $$prob' = \frac{1}{1+\exp(\Delta dom_{avg} / T)}$$

where $\Delta dom_{avg} = \frac{(\sum_{i=1}^{k} \Delta dom_{i,new}) + \Delta dom_{current,new}}{k}$

20. Set *selected solution* as *current solution* with *prob*
21. Set *new solution* as *current solution* with (1-*prob*)**prob'*
22. End if
23. Else if *new solution* dominates *current solution* /*Case 2b*/
24. Set *new solution* as *current solution*
25. Else if *new* and *current solution* are non-dominant to each other /*Case 2c*/
26. $$prob = \frac{1}{1+\exp(\Delta dom_{avg} / T)}$$

where $\Delta dom_{avg} = \frac{(\sum_{i=1}^{k} \Delta dom_{i,new})}{k}$

27. Set *new solution* as *current solution* with probability=*prob*
28. End if
29. Else if *new solution* and *Archive* are non-dominant to each other /*Case 3*/
30. Add *new solution* to the *Archive*
31. Set *new solution* as *current solution*
32. End if
33. End for
34. $T = \alpha * T$
35. End while

The inputs of MOSA/R for configuration optimization are the cylinders' position randomly initialized in the Cartesian coordinate system. MOSA/R uses the concept of the amount of domination in computing the acceptance probability of a new solution. Given two solutions **a** and **b**, the amount of domination is defined as

$$\Delta dom_{\mathbf{a},\mathbf{b}} = \prod_{i=1, f_i(\mathbf{a}) \neq f_i(\mathbf{b})}^{M} (\left| f_i(\mathbf{a}) - f_i(\mathbf{b}) \right| / R_i) \tag{2.20}$$

where M is the number of objectives and R_i is the range of the ith objective.

2.4.2 The re-seed scheme: when to trigger

The re-seed scheme (Case 2a-2) (Line 16 to Line 22) essentially differs MOSA/R from AMOSA. A re-seed scheme consists of two parts, "when" to re-seed and "what" to re-seed. "When" implies the scenarios that triggers the re-seed, which is problem independent. And "what" means if the re-seed

happens, which solution should be selected. Such selection process could be designed to accommodate different optimization needs. Let's first define when to re-seed in a general sense. (What to re-seed will be discussed in Section 2.5.3). Figure 2.8 illustrates when the re-seed may happen for both AMOSA and MOSA/R in the objective space and the decision space. Figures 2.8(a) and 2.8(b) depict the simplified scenarios where two objectives are considered while Figures 2.8(c) and 2.8(d) are the simplified scenarios with only one objective. In AMOSA, the probability re-seed happens when the *new solution* dominates *current solution*, and at the meantime, is dominated by at least one solution in the *Archive*. As illustrated in Figures 2.8(a) and 2.8(c), if re-seed takes place, a potential non-dominated solution (the dotted line circle) could be overlooked, which, as a consequence, will lead to premature convergence, and loss of diversity among the Pareto front. However, in MOSA/R, the probability re-seed happens when a *new solution* is dominated by both *current solution*, which does not belong to the *Archive*, and, at least, one solution in the *Archive*, as depicted in Figures 2.8(b) and 2.8(d). The re-seed timing of MOSA/R is improved compared to that of AMOSA in the sense that it gives the algorithm more flexibility to explore a possible optimal region. Figures 2.8(e) and 2.8(f) show that given the *current solution* and the *Archive*, how the optimizer performs correspondingly when the *new solution* falls into different quadrants for AMOSA and MOSA/R respectively. It is worth noting that a re-seed scheme is preferable for practical applications due to the fact that it will enable the algorithm to converge to the feasible region in a timely manner. A reasonable convergent time is significant because otherwise exhaustive random search, which could be easily proved to be convergent to optimal if given enough time, would be a panacea. Hence, even though Simulated Annealing has been proved to be convergent, the estimated time of absolute convergence $\left\|\boldsymbol{v}(kr)-\boldsymbol{e}^{*}\right\|=\mathrm{O}(1/k^{\min(a,b)})$ which is related to the objectives and annealing schedule (Mitra et al, 1986) is usually unrealistic. Under such circumstances, a properly designed re-seed could make a difference.

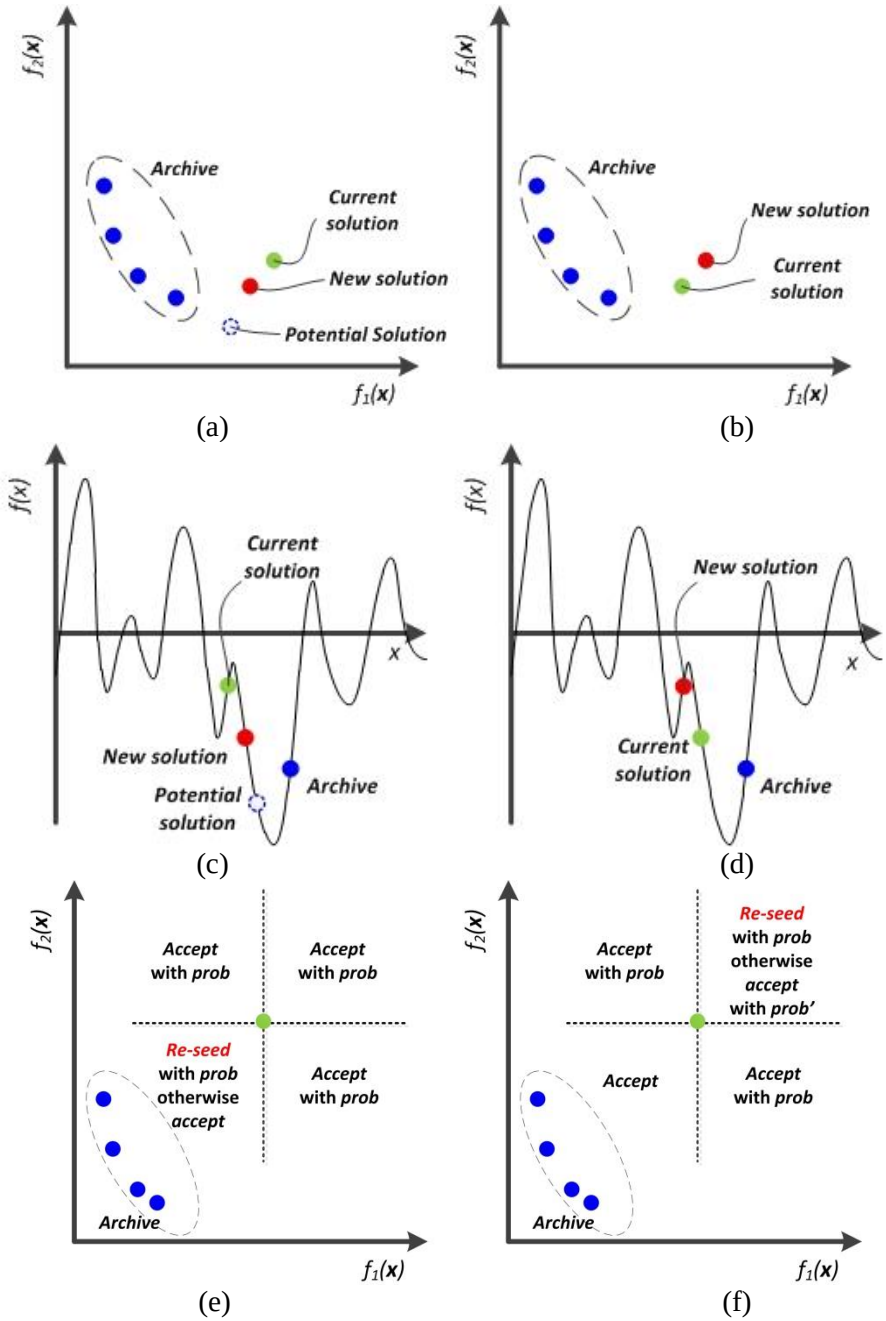


Figure 2.8 Re-seed timing comparisons of AMOSA (a) (c) (e) and MOSA/R (b) (d) (f).

The merit of simulated annealing is to allow inferior moves by probabilistic relaxation, so potentially an inferior solution will lead the search to a superior solution in the subsequent steps. The re-seed scheme should not weaken such merit. MOSA/R is designed followed the rule that once the new solution shows the potential for being the starting point of the very path towards the optimal region, re-seed may be triggered probabilistically. Table 4 gives a direct comparison of the decision procedure in MOSA/R and

AMOSA. The discussion of how to re-seed (i.e., Line 17 designed rules in the pseudo-code) is presented in the next section of implementation.

Table 2.4 Decision procedures of MOSA/R as compared to AMOSA

<table>
<tr><td colspan="2"></td><td colspan="2">MOSA/R</td><td>AMOSA</td></tr>
<tr><td rowspan="3">$New \prec Archive$
/*Case 1*/</td><td>$New \prec Cur$</td><td colspan="3" rowspan="2">Accept and Update</td></tr>
<tr><td>$New \cong Cur$</td></tr>
<tr><td>$New \succ Cur$</td><td colspan="3">Not possible</td></tr>
<tr><td rowspan="4">$New \succ Archive$
/*Case 2*/</td><td rowspan="2">$New \succ Cur$
/*2a*/</td><td>$Cur \in Archive$
/*2a-1*/</td><td>Accept with prob</td><td rowspan="2">Accept with prob</td></tr>
<tr><td>$Cur \notin Archive$
/*2a-2*/</td><td>Re-seed with prob
Accept with
(1- prob)*prob’</td></tr>
<tr><td>$New \prec Cur$
/*2b*/</td><td colspan="2">Accept</td><td>Re-seed with prob
Accept with (1- prob)</td></tr>
<tr><td>$New \cong Cur$
/*2b*/</td><td colspan="3">Accept with prob</td></tr>
<tr><td rowspan="3">$New \cong Archive$
/*Case 3*/</td><td>$New \prec Cur$</td><td colspan="3" rowspan="2">Accept and Update</td></tr>
<tr><td>$New \cong Cur$</td></tr>
<tr><td>$New \succ Cur$</td><td colspan="3">Not possible</td></tr>
</table>

2.5. Implementation details

2.5.1 Initialization

The algorithm begins with 100 initial solutions in the *Archive*. All these solutions, which correspond to possible configurations, are chosen randomly from the objective space. These solutions can also be refined using a simple hill-climbing technique. In this research, a randomly chosen initial *Archive* is employed to better exploit the performance of each algorithm.

2.5.2 Move routine

According to Line 6 of the pseudo-code, the new solution is obtained through perturbations. For the test problem, perturbation means to move the cylinders. In this research, two move routines are used, i.e., translation and rotation. To be more specific, translation perturbs x, y and z, and rotation perturbs θ and φ. The new perturbed decision variable is sampled from Laplace distribution whose expectation is the current solution and diversity parameters are chosen depending on the scale of each variable. The probability density function is given as

$$f(x \mid \mu, l) = \frac{1}{2l}\left(-\frac{|x-\mu|}{l}\right) \tag{2.21}$$

where μ denotes the expectation and l is the diversity parameter. For our case, l is chosen as 0.5 and 30 for translation and rotation, respectively. We only carry out one kind of perturbation to one cylinder per iteration for the purpose of detailed scanning of the objective space. Nevertheless, it is also viable to perturb multiple cylinders at once.

2.5.3 The re-seed scheme: what to pick

This subsection discusses what solution to select from the *Archive* (Line 17 designed rules) to compete for acceptance. One of the advantages of the selection process is that it could be adjusted flexibly with respect to different applications. In this research, two re-seed schemes are formulated. First is to select the solution from *Archive* that corresponds to the minimum difference of domination amount. Then the selected solution is set as a current solution with probability $1/(1+\exp(-\Delta dom_{\text{select,new}}))$. MOSA/R using the first approach is given the notation MOSA/R-1.0.

The second approach needs to make use of the technique called fast non-dominated sort (Deb et al, 2002). We sort the non-dominated solutions in *Archive* into fronts based on their objective functions that correspond to constraints (Equations (2.12), (2.13) and (2.14)). This is the step that embodies biased constraints handling. Then, the solution in the first front with a minimum amount of domination towards the new solution is chosen and will be set as the current solution with probability $1/(1+\exp(-\Delta dom_{\text{select,new}}))$. MOSA/R using the second approach is given the notation MOSA/R-2.0. In both approaches, the solution corresponding to the minimum difference of domination amount is chosen instead of the solution corresponding to the maximum difference of domination to avoid premature convergence. Figure 2.9 illustrates how to choose one solution among *Archive* as a re-seed candidate when there are two penalties. In our case study, most solutions along the process are infeasible solutions, so the issue of maintaining diversity among population seems somewhat trivial. When a well distributed and well spread non-dominated set is something needs to be achieved, techniques such as fitness sharing

(Goldberg and Richardson, 1987), isolation by distance (Ryan, 1995), over specification, crowding (Deb et al, 2002), ε-dominance (Laumanns et al, 2002), and decomposition (Zhang and Li, 2007) could be incorporated into the selection process. Take decomposition, for example. Whenever the re-seed is triggered, solutions in *Archive* should be converted into scalars with a different emphasis from time to time. Then the solution that is closest to the new solution in terms of the scalar value will be selected to compete for acceptance.

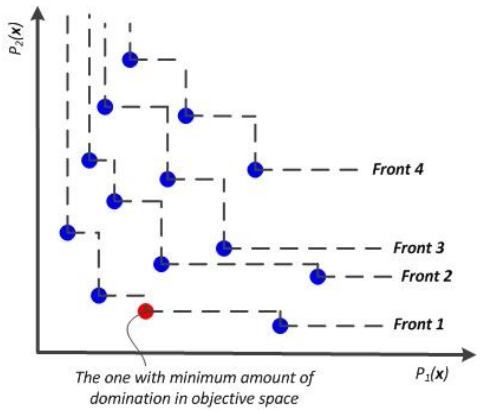


Figure 2.9 How to re-seed when constraints are considered.

2.5.4 Annealing schedule

The initial temperature is determined in such a way that virtually all transitions are accepted at the beginning 'burn in' period. Let the error be defined as

$$e_r = |(f^* - f_{\text{opt}}) / f_{\text{opt}}| \tag{2.22}$$

where f^* is the current objective value, and f_{opt} is the desired objective value. The stopping criteria, i.e., the final temperature should be chosen either to control the error defined above. In this research, for cases presented in Sections 2.6.1 and 2.6.2 the starting temperature $T_{\max}$ and final temperature $T_{\min}$ values are set to be 100 and 10^{-4}. For the test problem analyzed in Section 2.6.3, the starting temperature $T_{\max}$ and final temperature $T_{\min}$ values are set to be 1000 and 10^{-2}, respectively. Another parameter of an annealing schedule is the number of iterations, denoted as *iter*, performed at each temperature. It should be chosen such that the system is sufficiently close to the stationary distribution at that temperature. In this study,

iter between two consecutive temperature levels are specified as 81, 162 and 200 for cases analyzed in Sections 2.6.1, 2. 6.2 and 2.6.3, respectively.

For temperature decrement $T = \Phi(T)$, a common parameter to control the decrement is

$$T_{i+1} = \alpha T_i \tag{2.23}$$

where $0 < \alpha < 1$ is a constant. Some other cooling schedules available in the literature are logarithmic, Cauchy, and exponential. It has been proved that if annealed sufficiently slowly, SA converges to the global optimum. To have a fair comparison and enable a good exploitation of the solution space, the cooling rate of lowering temperature value is set to be 0.8 for cases analyzed in Sections 2.6.1 and 2.6.2, and 0.95 for the test problem analyzed in Section 2.6.3.

2.5.5 Implementation of other algorithms for comparison

In order to perform fair comparisons, each parameter in AMOSA is set to be the same as that of MOSA/R. For NSGA-II and MOEA/D, the total number of function evaluations is set in accordance with AMOSA and MOSA/R. Other parameters used follow the original publications of the algorithms. The population size is set to be 300. The distribution indexes in SBX and the polynomial mutation are set to be 20. The crossover rate is 1.00 and the mutation ration is 1/n where n is length of decision vector. In MOEA/D, Tchebycheff approach is used and the size of neighbor population is set to be 20. All initial solutions are generated randomly form the decision space of the problems.

2.6. Case Investigations

In this section, we first examine the performance of MOSA/R using benchmark multi-objective constrained problems as compared to MOEA/D, NAGA-II and AMOSA, and then apply it to the constraint configuration optimization problem defined in Section 2.2. The simulation results are based on 10 independent test runs. All simulations are carried out within MATLAB on a 2.40GHz Xeon E5620 computer.

For multi-objective optimization (MOO), an algorithm will provide a set of solutions that realize the optimal trade-offs between the considered optimization objectives, i.e., Pareto set. Therefore, the

comparison of the performance of MOO algorithms has to be based on their Pareto sets. There are multiple optimization goals in our multi-objective optimization: 1) number of feasible solutions; 2) convergence to the true Pareto-optimal set; and 3) maintenance of diversity in solutions of the Pareto-optimal set. Hence, six metrics are briefly explained below, and will be used in Sections 2.6.1, 2.6.2 and 2.6.3 to measure the results in terms of the above-mentioned goals.

Cardinality (N)

N is the number of feasible solutions in the Pareto set. For constrained multi-objective optimization problems, finding feasible region is a difficult task of itself. *Cardinality* (*N*) reflects how good the algorithm is in satisfying constraints and exploring feasible search space.

Inverted Generational Distance (IGD)

The *IGD* indicator quantifies the degree of convergence by computing the average of the minimum distance of points in the true Pareto front (*PF**) to points in Pareto front obtained (*PF*), as expressed below,

$$IGD(PF, PF^*) = \frac{\sum_{\mathbf{f}^* \in PF^*,\ i=1}^{|PF^*|} \sqrt{\min_{\mathbf{f} \in PF} \left(\sum_{m=1}^{M} (f_m^i - f_m)^2 \right)}}{|PF^*|} \tag{2.24}$$

where M is the number of objectives, f_m is the mth objective value of $\mathbf{f} \in PF$. In Equation (2.24), $\min_{\mathbf{f} \in PF} \left(\sum_{m=1}^{M} (f_m^i - f_m)^2 \right)$ calculates the minimum Euclidean distance between the ith point in *PF** and points in *PF*. A lower value of *IGD* indicates better convergence and completeness of the *PF* obtained.

HV indicator

The HV indicator measures convergence as well as diversity. Specifically,

$$HV(PF, r^*) = volume\left(\bigcup_{x \in PF} v(x, r^*) \right) \tag{2.25}$$

where *r** is the reference point which is set to be 1.1 times the upper bound of the Pareto front in the *HV* calculation. The calculation of *HV* requires normalized objective function values. In this research *HV*

stands for the percentage covered by the Pareto front of the cuboid defined by the reference point and the origin *(0, 0, 0)*.

Convergence of two sets (C)

When the true Pareto-optimal set is not available, the relative goodness of two sets of solutions should be measured. Let **A**, **B** be two approximation sets consisting of decision vectors. The function *C* maps the ordered pair (**A**, **B**) to the interval [0, 1] in the following manner

$$C(\mathbf{A},\mathbf{B}) = \frac{|\{\mathbf{b} \in \mathbf{B} \mid \exists \mathbf{a} \in \mathbf{A} : \mathbf{a} \prec \mathbf{b}\}|}{|\mathbf{B}|} \tag{2.26}$$

The value $C(\mathbf{A},\mathbf{B}) = 1$ means that all solutions in **B** are dominated by **A**. If $C(\mathbf{A},\mathbf{B}) = 0$ then no point in **B** is dominated by points in **A**. Both directions have to be considered for the comparison since $C(\mathbf{A},\mathbf{B})$ is not necessarily equal to $1 - C(\mathbf{B},\mathbf{A})$. As the true Pareto set is unknown, the measure *C* is used for the performance comparison of two multi-objective algorithms.

We also define that when **B** is empty while **A** is not, $C(\mathbf{A},\mathbf{B}) = 1$ and $C(\mathbf{B},\mathbf{A}) = 0$. When **A**, **B** are both empty, $C(\mathbf{A},\mathbf{B}) = C(\mathbf{B},\mathbf{A}) = 0.5$.

Minimal Spacing (S_m)

Spacing concerns the distribution of decision vectors throughout the non-dominated solutions found. A metric measuring the range (distance) variance of neighboring vectors in the Pareto set called *Minimal Spacing* (*S*) has been proposed (Srinivas and Deb, 1994). For a general *M*-dimensional problem, the *Spacing* metric is defined as:

$$S = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\bar{d} - d_i)^2} \quad (i = 1, 2, ..., N) \tag{2.27}$$

where

$$d_i = \min_j \left(\sum_{m=1}^{M} | f_m^i(x) - f_m^j(x) |\right), i \neq j \tag{2.28}$$

Then the average $\bar{d}$ of these distances is calculated, and *N* is the number of solutions on the non-dominated front. A value of zero for this metric indicates all members of the Pareto front currently

available are equidistantly spaced. The *Spacing* metric, however, may fail under common scenarios. Alternatively, *Minimal Spacing* (S_m) can be formulated based on *Spacing (S)* following the steps in (Bandyopadhyay et al, 2004). In brief, when computing d_i, the term $|f_m^i(x) - f_m^j(x)|$ is divided by $|F_m^{\max} - F_m^{\min}|$ in order to standardize it, where $F_m^{\max}$ and $F_m^{\min}$ are the maximum and minimum values respectively of the *m*th objective.

Accounted Proportion P

In this research, another metric called *Accounted Proportion* is formulated to compare the solutions obtained by several optimization algorithms over multiple simulations at once. Suppose K (≥ 2) algorithms are applied to the optimization problem, and the simulation is carried out L times for each algorithm respectively. PS_i^j represents the Pareto set obtained by algorithm i in its jth simulation. Let the non-dominated solutions of the union of Pareto sets obtained by algorithm i over L simulations be PS_i^*,

$$PS_i^* = \text{Non-dominated}\left(\bigcup_{j \in L} \{PS_i^j\}\right) \tag{2.29}$$

And the set of non-dominated solutions of all Pareto sets is PS^*.

$$PS^* = \text{Non-dominated}\left(\bigcup_{i \in K,\ j \in L} \{PS_i^j\}\right) \tag{2.30}$$

Thus for algorithm i, the *Accounted Proportion* is

$$P_i = |PS_i^*| / |PS^*| \tag{2.31}$$

where $|\cdot|$ operation calculates the number of solutions in a set.

2.6.1 Case 1: SRN problem

We first examine the general performance of MOSA/R by applying five algorithms (MOEA/D, NSGA-II, AMOSA, MOSA/R-1.0, MOSA/R-2.0) to a constrained multi-objective benchmark problem SRN (Srinivas and Deb, 1994):

Minimize: $f_1(\mathbf{x}) = 2 + (x_1 - 2)^2 + (x_2 - 2)^2$

$f_2(\mathbf{x}) = 9x_1 - (x_2 - 1)^2$

subject to: $x_1^2 + x_2^2 \le 225$

$x_1 - 3x_2 + 10 \le 0$

$-20 < x_1, x_2 < 20$ (2.32a-f)

The simulation results are presented in Table 5 where three metrics, *Cardinality* and *IGD* and *HV*, are used. The shaded grids in Table 2.5 indicate the best result in each test in terms of the measures. As demonstrated in Figure 10, MOSA/R-2.0 prevails in *Cardinality* and *IGD*, and is the second best in terms of *HV*. Meanwhile, MOSA/R-1.0 shows better performance than the other three algorithms in *Caridnality* and *IGD*, and reach the third place in *HV* next to NSGA-II and MOSA/R-2.0. In this case, it is relatively easier for each algorithm to find feasible solutions. As shown in Figure 2.11, all algorithms applied are able to locate the true Pareto front in limited 5,000 function evaluations, but MOSA/R-2.0 exhibits the best coverage over the true Pareto front.

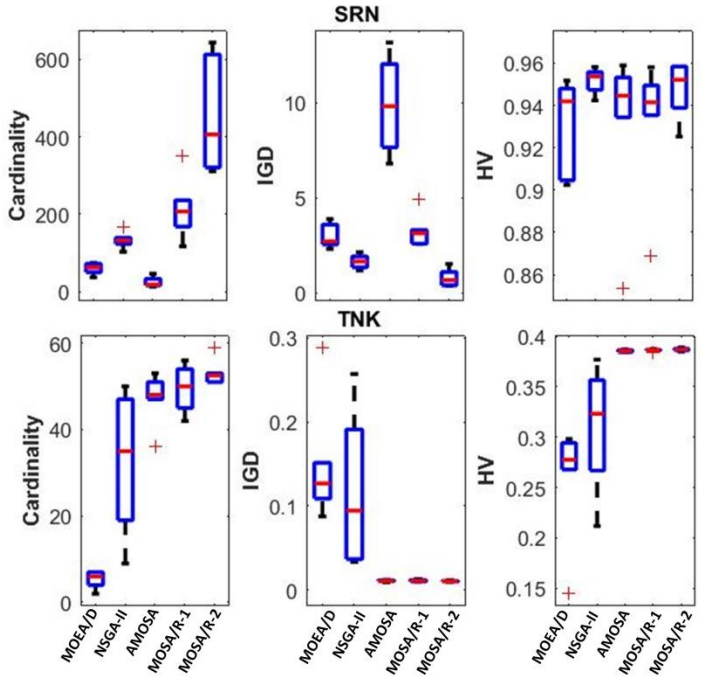


Figure 2.10 Box plots of five algorithms w.r.t. *Cardinality*, *IGD* and *HV* on SRN and TNK (From left to right: MOEA/D, NAGA-II, AMOSA, MOSA/R-1.0, and MOSA/R-2.0)

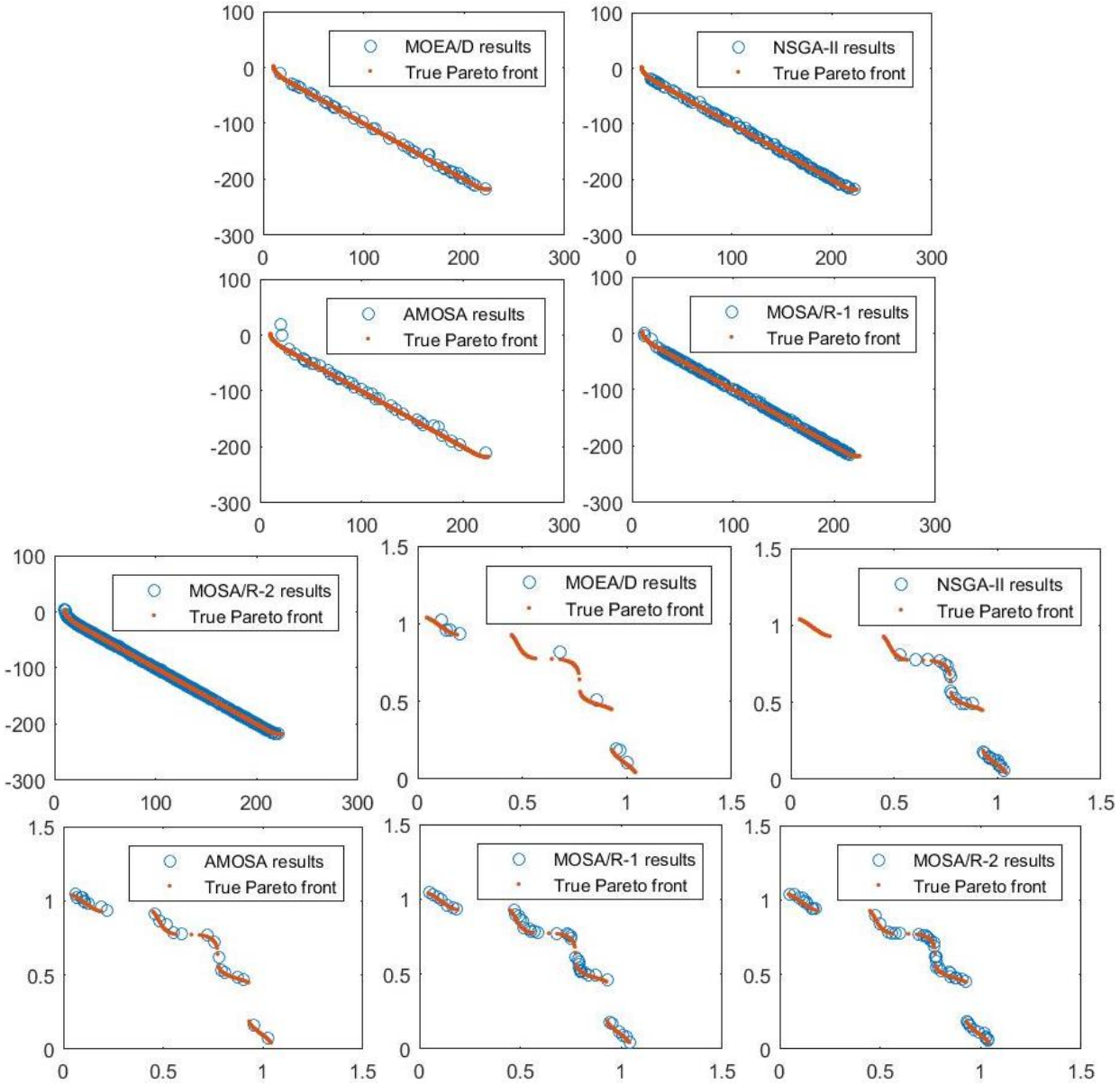


Figure 2.11 Pareto front obtained by each algorithm for test instances SRN and TNK (First 5 SRN, last 5 TNK)

2.6.2 Case 2: TNK problem

Next the algorithms are applied to a more difficult benchmark constrained multi-objective problem TNK (Tanaka et al, 1995). The problem is given as follows:

Minimize: $f_1(\mathbf{x}) = x_1$

$f_2(\mathbf{x}) = x_2$

Subject to: $$x_1^2 + x_2^2 \geq 1 + 0.1\cos(16\arctan\frac{x_2}{x_1})$$

$$(x_1 - 0.5)^2 + (x_2 - 0.5)^2 \leq 0.5$$

$$0 < x_1, x_2 < \pi \qquad \text{(2.33a-f)}$$

Equation (2.33f) is modified by enlarging the design space to $0 < x_1, x_2 < 100$ in order to make it more challenging in finding feasible solutions. Compared to SRN, it is harder to locate the feasible region because this problem has larger decision space and the true Pareto front is discrete and non-linear. For each algorithm, it runs until 10,000 function evaluations are reached. The simulation results are presented in Table 2.5. As demonstrated in Figure 10, three Simulated Annealing based algorithms have better performance than Genetic algorithms, which validates our discussion in Section 2.4. MOSA/R-2.0 has the best overall performance among the five algorithms. Consistent with the case evaluated in Section 6.1, both MOSA/R algorithms achieve more feasible solutions than MOEA/D, NSGA-II and AMOSA. In addition, as Figure 2.11 illustrates, both MOSA/R maintain relatively good diversity compared to other algorithms.

Table 2.5 *Cardinality*, *IGD* and *HV* measures on the constrained benchmark tests SRN and TNK over 10 runs in terms of mean

Algorithm	SRN			TNK		
	Cardinality	*IGD*	*HV*	*Cardinality*	*IGD*	*HV*
MOEA/D	59.6667	2.9692	0.9318	5.3333	0.14838	0.25995
NSGA-II	132.5	1.6596	0.9519	32.5	0.1179	0.3097
AMOSA	24	9.89	0.93155	47.16667	0.010917	0.38555
MOSA/R-1.0	214.3333	3.2863	0.9325	49.5	0.0113	0.3861
MOSA/R-2.0	450.1667	0.7966	0.9476	53.1667	0.010633	0.3869

In both cases illustrated in Sections 2.6.1 and 2.6.2, MOSA/R has an edge over some other contemporary approaches when applied to constrained multi-objective benchmark problems. Given limited number of function evaluations, MOSA/R converges faster and yields more useful feasible results.

2.6.3 Case 3: Configuration optimization test problem

In this sub-section, we will solve the configuration optimization test problem defined in Section 2. For this problem, there are three main goals that an MOO algorithm should reach. It should find as many feasible solutions as possible. It should converge as close to the true Pareto front as possible. It should also maintain as diverse a solution set as possible. As mentioned earlier in Section 2.6, the comparison metrics *N* shows the number of feasible solutions (the larger the better), *C* represents the degree of convergence with respect to the compared set of solution (the closer to 1 the better), *P* illustrates how well the algorithm converges from a different perspective when the strategies applied are considered all at once (the larger the better), and S_m indicates the diversity (the smaller the better).

The numerical experiment is designed in such a way that we decrease one parameter *SL* (i.e., tightening the constraints), which is the length of cube side, from 9.4 to 8.2 with a step of size 0.1. The cut-off *SL* value is chosen to be 8.2 because this is when AMOSA have difficulties finding feasible solutions. For each *SL* value, AMOSA, MOSA/R-1.0 and MOSA/R-2.0 are devised. All algorithms are executed 10 times, and the results reported are the mean values and standard deviations obtained over 10 runs. Genetic algorithms (MOEA/D and NSGA-II) are not included in the comparison because they are not able to find feasible solutions. The possible reasons have been discussed in Section 2.4. Notations AM and MR in the tables mentioned below stand for AMOSA and MOSA/R respectively.

Tables 2.6 and 2.7 demonstrate the relative performance of AMOSA, MOSA/R-1.0, and MOSA/R-2.0 in terms of the three metrics *N*, *C* and S_m. The comparisons are also illustrated in Figures 2.12 and 2.13. The number of feasible solutions (*Cardinality*) found by MOSA/R-1.0 and MOSA/R-2.0 are always greater than that of AMOSA, as can be seen from Figure 2.12(a), and MOSA/R-2.0 has a clear advantage in this contest because the re-seed scheme of MOSA/R-2.0 is designed for finding feasible solutions under hard constraints. When AMOSA and MOSA/R-1.0 start to show no feasible solution as constraints are tightening, MOSA/R-2.0 excels at *Cardinality* considerably.

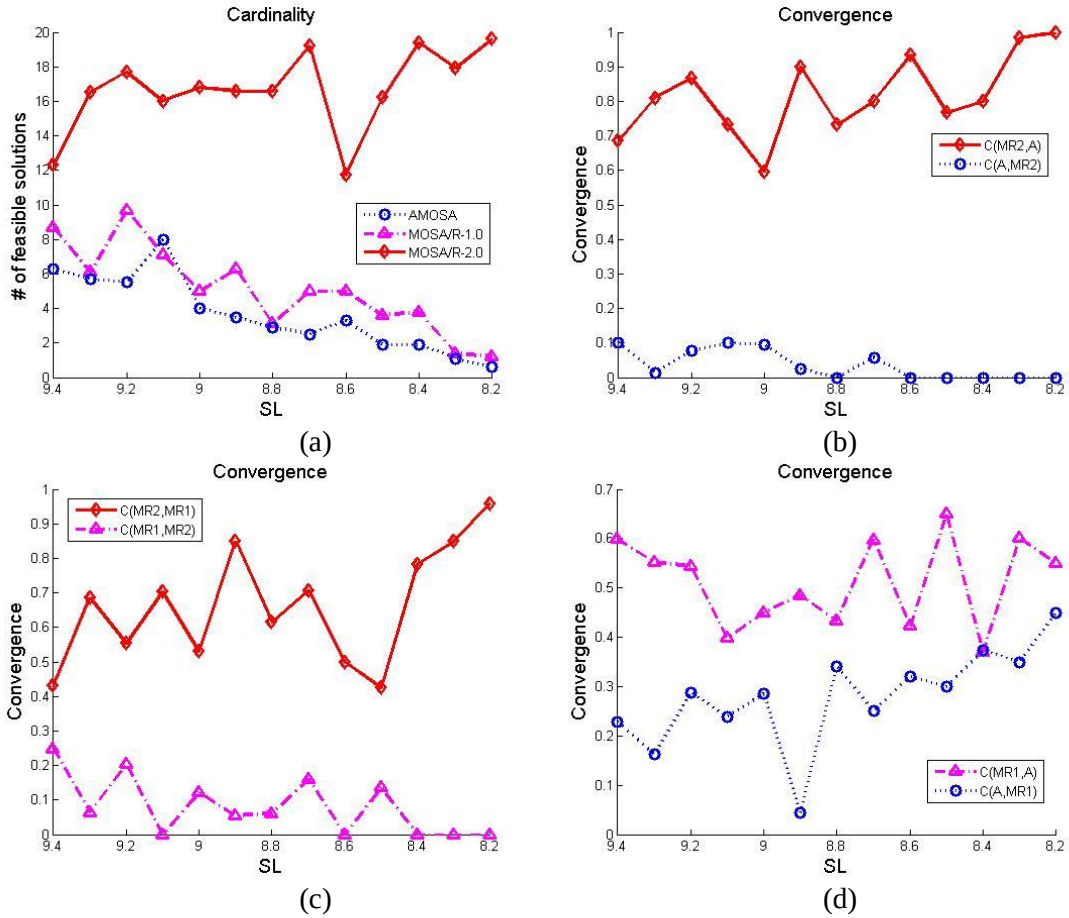


Figure 2.12 *Cardinality* and *Convergence* with *SL* varying from 9.4 to 8.2.

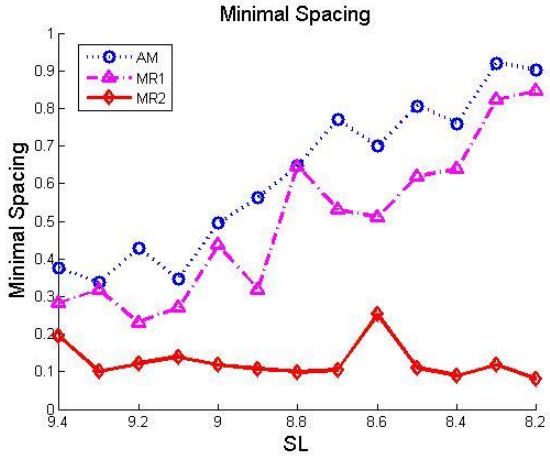


Figure 2.13 *Minimal Spacing* with *SL* varying from 9.4 to 8.2.

Table 2.6 *Cardinality* and *Minimal Spacing* measures on the configuration optimization test problem over 10 runs in terms of mean and standard deviation

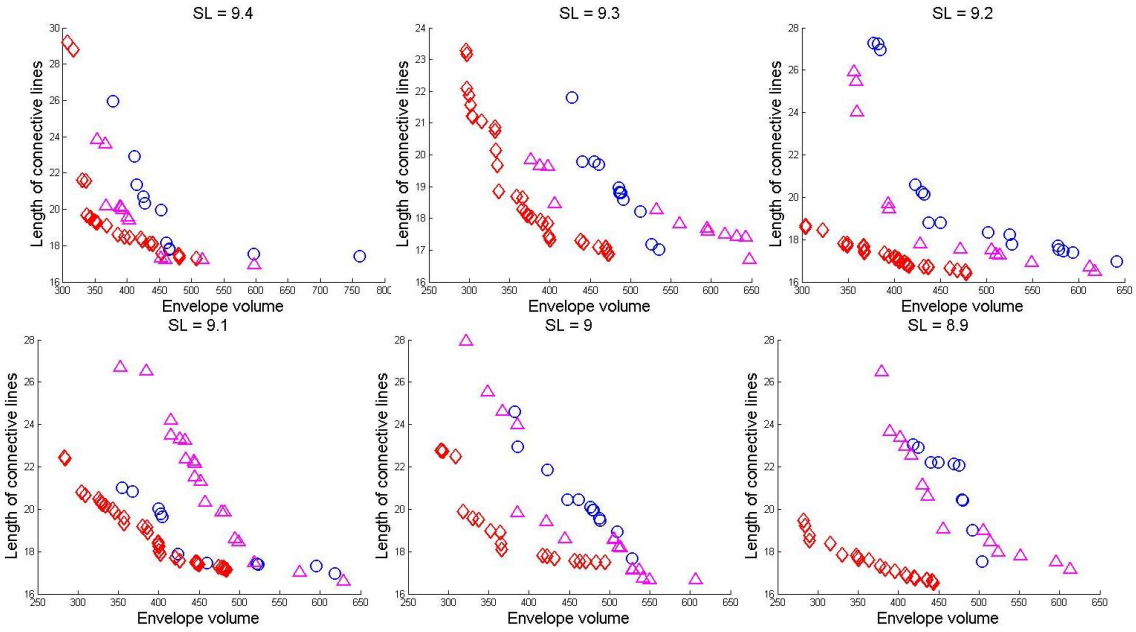
SL	*Cardinality (N)*			*Minimal Spacing (S_m)*		
	AM	MR-1.0	MR-2.0	AM	MR-1.0	MR-2.0
9.4	6.3 (2.87)	8.7 (3.37)	12.3 (5.23)	0.3756 (0.23)	0.2822 (0.13)	0.1951 (0.08)

9.3	5.7 (2.40)	6.1 (2.92)	16.5 (6.93)	0.3368 (0.28)	0.3177 (0.26)	0.1003 (0.04)
9.2	5.5 (3.54)	9.7 (4.06)	17.7 (7.73)	0.4279 (0.40)	0.2298 (0.11)	0.1218 (0.06)
9.1	8.0 (4.71)	7.1 (3.73)	16.0 (8.65)	0.3471 (0.35)	0.2693 (0.14)	0.1398 (0.13)
9.0	4.0 (2.40)	5.0 (3.09)	16.8 (6.66)	0.4946 (0.36)	0.4359 (0.25)	0.1178 (0.05)
8.9	3.5 (1.35)	6.3 (2.63)	16.6 (5.60)	0.5618 (0.33)	0.3182 (0.26)	0.1078 (0.04)
8.8	2.9 (2.56)	3.1 (1.79)	16.6 (6.02)	0.6469 (0.33)	0.6457 (0.39)	0.0994 (0.04)
8.7	2.5 (2.68)	5.0 (3.53)	19.2 (8.35)	0.7693 (0.37)	0.5301 (0.34)	0.1041 (0.05)
8.6	3.3 (2.26)	5.0 (3.56)	11.7 (8.67)	0.6990 (0.27)	0.5112 (0.35)	0.2526 (0.28)
8.5	1.9 (1.20)	3.6 (2.99)	16.2 (8.04)	0.8069 (0.31)	0.6194 (0.36)	0.1099 (0.06)
8.4	1.9 (2.18)	3.8 (5.01)	19.4 (10.85)	0.7602 (0.39)	0.6384 (0.40)	0.0900 (0.05)
8.3	1.1 (1.85)	1.4 (1.35)	17.9 (9.16)	0.9199 (0.25)	0.8237 (0.29)	0.1181 (0.13)
8.2	0.6 (1.08)	1.2 (2.15)	19.6 (7.30)	0.9023 (0.31)	0.8458 (0.33)	0.0798 (0.03)

Table 2.7 *Convergence* measure on the configuration optimization test problem over 10 runs in terms of mean and standard deviation

SL	*Convergence (C)*					
	C(MR1,AM)	C(AM,MR1)	C(MR1,MR2)	C(MR2,MR1)	C(MR2,AM)	C(AM,MR2)
9.4	0.5985 (0.43)	0.2279 (0.34)	0.2474 (0.42)	0.4303 (0.45)	0.6845 (0.39)	0.1000 (0.32)
9.3	0.5525 (0.35)	0.1625 (0.35)	0.0633 (0.13)	0.6848 (0.40)	0.8092 (0.28)	0.0143 (0.05)
9.2	0.5443 (0.49)	0.2883 (0.41)	0.2042 (0.36)	0.5533 (0.48)	0.8667 (0.32)	0.0765 (0.24)
9.1	0.3980 (0.37)	0.2387 (0.35)	0 (0)	0.7031 (0.30)	0.7317 (0.38)	0.1000 (0.32)
9.0	0.4500 (0.48)	0.2857 (0.46)	0.1209 (0.16)	0.5307 (0.43)	0.5958 (0.47)	0.0944 (0.20)
8.9	0.4850 (0.45)	0.0433 (0.11)	0.0545 (0.12)	0.8494 (0.29)	0.9000 (0.32)	0.0241 (0.08)
8.8	0.4333 (0.50)	0.3400 (0.47)	0.0600 (0.19)	0.6150 (0.46)	0.7333 (0.41)	0 (0)
8.7	0.5972 (0.45)	0.2500 (0.42)	0.1597 (0.32)	0.7067 (0.42)	0.8000 (0.42)	0.0580 (0.14)
8.6	0.4226 (0.46)	0.3200 (0.42)	0 (0)	0.5000 (0.48)	0.9333 (0.14)	0 (0)
8.5	0.6500 (0.47)	0.3000 (0.48)	0.1352 (0.24)	0.4250 (0.50)	0.7667 (0.42)	0 (0)
8.4	0.3700 (0.40)	0.3733 (0.40)	0 (0)	0.7833 (0.34)	0.8000 (0.42)	0 (0)
8.3	0.6000 (0.46)	0.3500 (0.41)	0 (0)	0.8500 (0.34)	0.9833 (0.05)	0 (0)
8.2	0.5500 (0.37)	0.4500 (0.37)	0 (0)	0.9583 (0.09)	1.0000 (0)	0 (0)

MOSA/R-2.0 is the best among three in the regard of *Convergence* as well. Meanwhile, MOSA/R-1.0 provides generally better performance than AMOSA. The exception happens when *SL* equals to 8.4 (Figure 2.12(d)) which may not be the case when more simulations are executed. The reason for such fluctuation is that all three algorithms are heuristic that do not guarantee to find the exact optimal solutions. However, they are capable of providing immediate results that are very close to optimality efficiently and handle optimization problems that are currently out of the reach of theoretically rigorous methodology. As a result, the quality of solutions of heuristic methods has no robust justification. The *Minimal Spacing* curve of MOSA/R-2.0 remains low, indicating that it maintains better diversity than MOSA/R-1.0 and AMOSA. MOSA/R-1.0 is no worse than AMOSA at *Minimal Spacing* in all tested *SL*s. We should note here that for the test problem, a set of well-spread solutions doesn't necessary mean it is superior to that with relatively bad diversity. Nevertheless, because a set of well-spread solutions usually belong to the same class of configuration, a set of ill-distributed solutions may belong to different classes of configuration meaning possibly better diversity.



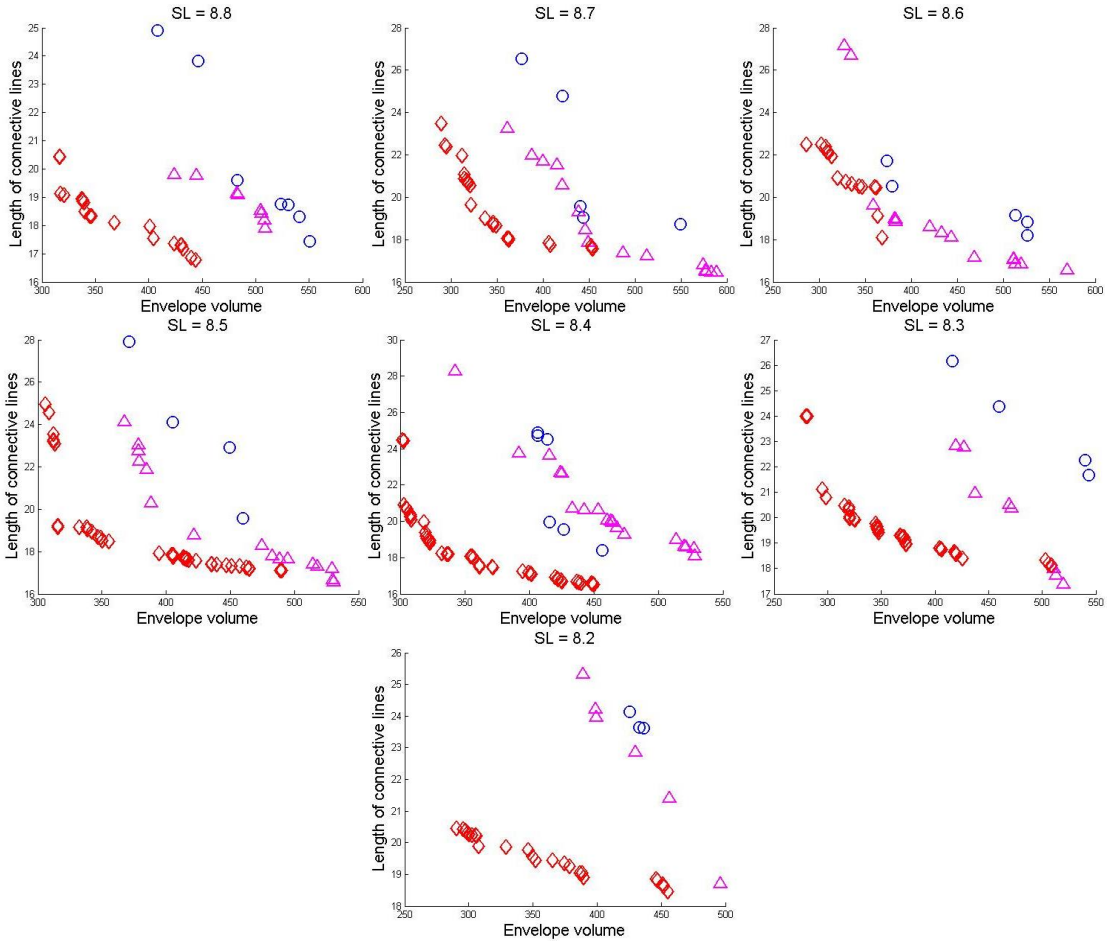
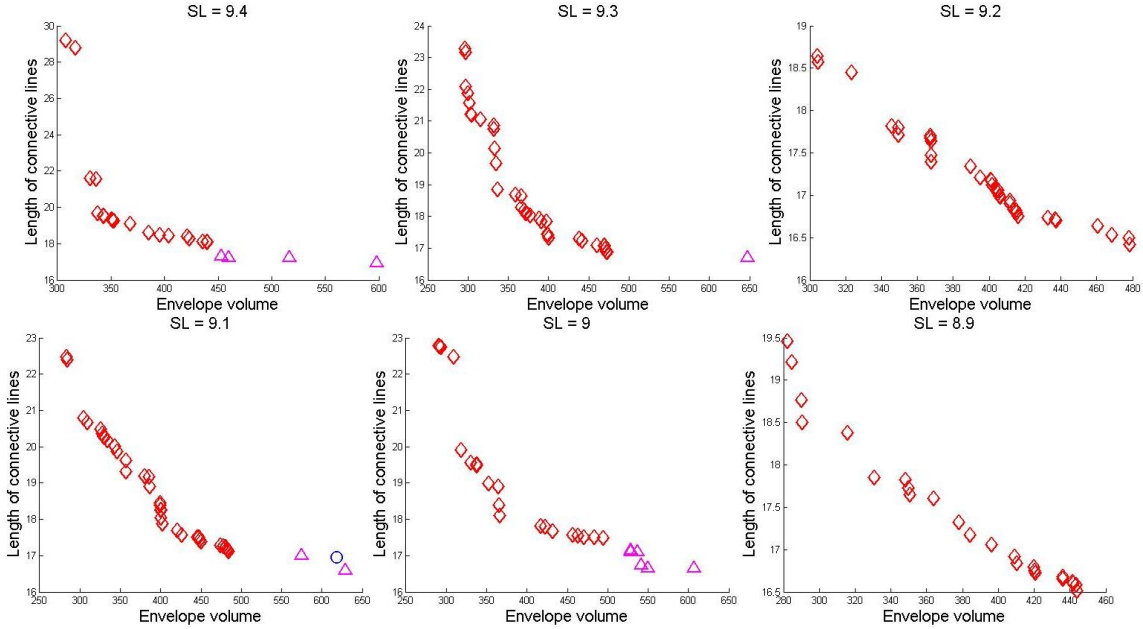


Figure 2.14 Pareto set after 10 test runs of the three algorithms as *SL* varying from 9.4 to 8.2 (◊: MOSA/R-2.0; Δ: MOSA/R-1.0; O: AMOSA).



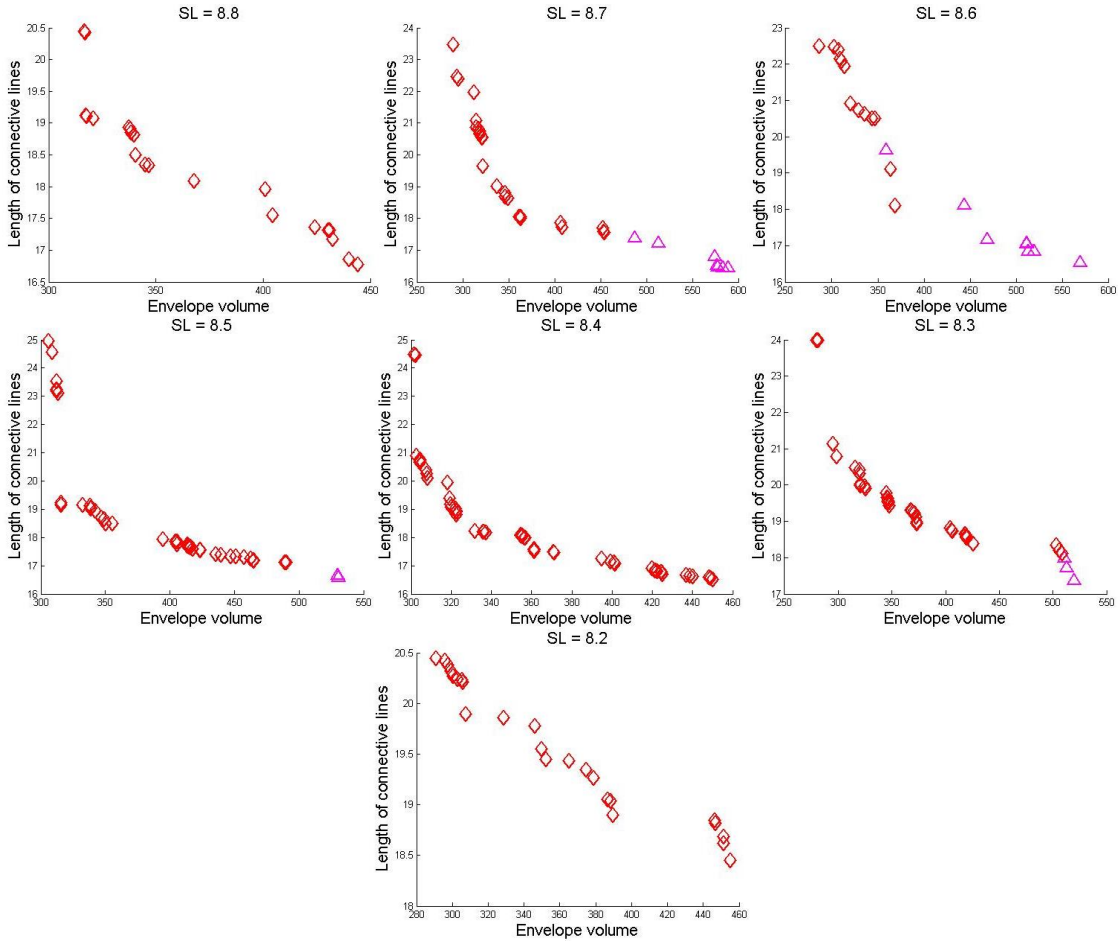


Figure 2.15 Best combined Pareto set over 10 runs as *SL* varying from 9.4 to 8.2 (◊: MOSA/R-2.0; Δ: MOSA/R-1.0; O: AMOSA).

Figure 2.14 depicts the best solutions found in 10 test runs by the three algorithms respectively as *SL* varies from 9.4 to 8.2. Figure 2.15 shows the combined best Pareto set over 10 runs for each *SL*. The *Accounted Proportion* for each algorithm is calculated and shown in Table 2.8. Once again, MOSA/R-2.0 prevails, and MOSA/R-1.0 is the second best. Figure 2.16 depicts average *Accounted Proportion* of each algorithm considering all simulations.

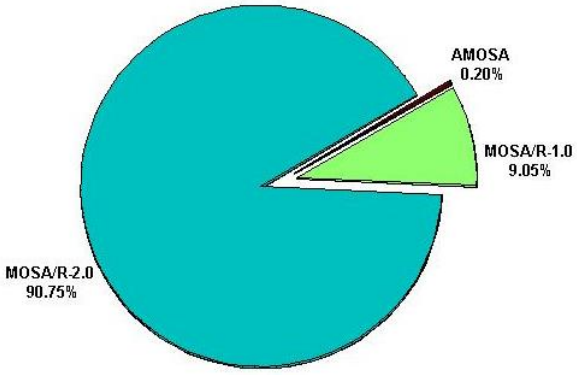


Figure 2.16 Average *Accounted Proportion* of each algorithm.

Table 2.8 *Accounted Proportion* of solutions obtained by each algorithm in the combined Pareto sets

SL	*Accounted Proportion (P)*		
	AMOSA	MOSA/R-1.0	MOSA/R-2.0
9.4	0%	16.00%	84.00%
9.3	0%	2.86%	97.14%
9.2	0%	0%	100%
9.1	2.63%	5.26%	92.11%
9.0	0%	22.22%	77.78%
8.9	0%	0%	100%
8.8	0%	0%	100%
8.7	0%	21.88%	78.13%
8.6	0%	38.10%	61.90%
8.5	0%	4.17%	95.83%
8.4	0%	0%	100%
8.3	0%	7.14%	92.86%
8.2	0%	0%	100%
Average	0.2023%	9.0485%	90.7500%

The numerical results illustrate that MOSA/R has desirable overall performance in terms of *Cardinality*, *Convergence*, *Minimal Spacing*, and *Accounted Proportion* for the test problem as the constrained level elevates. To be more specific, MOSA/R-2.0 is far ahead of AMOSA in all four contests while MOSA/R-1.0, which could be considered as a general version of MOSA/R-2.0 or an improved version of AMOSA, has an edge over AMOSA. In this research, due to length limitations, only the interval where AMOSA starts having difficulties finding feasible solutions is considered in order to demonstrate the capability of MOSA/R under such situation. Videos that illustrate the optimization

process of MOSA/R-1.0 and MOSA/R-2.0 applied to the configuration optimization test problem when *SL* is 8.7 can be found in https://youtu.be/zmdwNsyZYow and https://youtu.be/YeHrtSHY5ss respectively. As demonstrated in the first video, MOSA/R-1.0 locates the first feasible solution at temperature 32.1723 and that is the only feasible solution it is capable of finding. On the other hand, as shown in the second video, MOSA/R-2.0 finds the first feasible solution at temperature 440.1267, and finally provides the user with 32 feasible solutions. Notice in the second video, the current solution frequently jumps back to the archived solutions that when re-seed happens.

2.7. Conclusion

This research provides a systematic framework of solving configuration optimization problems in additive manufacturing. In particular, the envelope volume in space occupied by a number of components is to be minimized along with minimizing the length of connectivity lines between the components under multiple hard constraints. For this class of problems, the optimizer is required not only to locate the feasible regions, but also to maintain convergence and diversity among the feasible solutions. We develop an improved MOSA algorithm (MOSA/R) towards MOO problems, featuring a newly designed re-seed scheme that is able to redirect the optimizer towards the regions of better feasibility and optimality. Two versions of MOSA/R are formulated, i.e., the general version MOSA/R-1.0 and a version for constrained problems MOSA/R-2.0. For benchmark constrained MOO problems, MOSA/R shows better performance than MOEA/D, NSGA-II and AMOSA. For the constrained configuration optimization problem, it is identified that when all methods are applicable, MOSA/R-2.0 yields better results than MOSA-1.0 and AMOSA, and MOSA/R-1.0 has better performance than AMOSA in general senses. This new framework can be employed to facilitate an integrated process of design automation and manufacturing optimization. The re-seed scheme of MOSA/R could potentially be designed to meet different needs that prioritize, for example, distribution or diversity. The investigations of different re-seed schemes will be carried out in our future works. Meanwhile, convex-like optimization problems are of interest rather than multi-modal problems, MOSA/R, as a stochastic technique, could be slower than

deterministic techniques such as Pattern Search or DIRECT method. Therefore, it is more preferable to use deterministic approaches to solve convex/concave optimization problems. The MOSA/R scheme is also a good fit for general multi-modal optimization problems that have multiple objectives and constrains.

Chapter 3. Design for Additive Manufacturing: Optimization of Piping Network in Compact System with Enhanced Path-Finding Approach

This research aims at unleashing the potential of additive manufacturing technology in industrial design that can produce structures/devices with irregular component geometries to reduce sizes/weights. We explore, by means of path-finding, the length minimization of freeform hydraulic piping network in compact space under given constraints. Previous studies on path-finding have mainly focused on enhancing computational efficiency due to the need to produce rapid results in such as navigation and video-game applications. In this research, we develop a new Focal Any-Angle A* approach that combines the merits of grid-based method and visibility graph-based method. Specifically, we formulate pruned visibility graphs preserving only the useful portion of the vertices, and then find the optimal path based on the candidate vertices using A*. The reduced visibility graphs enable us to outperform approximations and maintain the optimality of exact algorithms in a more efficient manner. The algorithm proposed is compared to the traditional A* on Grids, Theta* and A* on Visibility Graphs in terms of path length, number of nodes evaluated, as well as computational time. As demonstrated and validated through case studies, the proposed method is capable of finding the shortest path with tractable computational cost, which provides a viable design tool for the additive manufacturing of piping network systems.

3.1 Introduction

The recently rapid development of additive manufacturing technology has induced paradigm-shifting advancements on several fronts. A salient feature of additive manufacturing is its capability to produce structures/devices with irregular and freeform geometries that otherwise cannot be realized by traditional manufacturing technologies (Gibson et al., 2010; Frazier, 2014; Zhou, 2014) due to the difficulties in parts fabrication and assemblage. Therefore, there exists unprecedented opportunities to facilitate optimal designs of structures/devices with significantly reduced sizes/weights while maintaining or even improving their functional performances. One of such opportunities, which has not been thoroughly

investigated, is to minimize the lengths/weights of various networking components, e.g., hydraulic piping, in vehicular, marine, and airborne systems where the reduction of sizes and weights of these systems has significant payoff (Cao et al, 2016). As shown in Figure 3.1(a) and 3.1(b), limited by tooling capability, a connecting pipeline designed for traditional manufacturing consists of only straight line segments and machining accesses, whereas the pipelines design for additive manufacturing could be free-form curves of shorter length, as exemplified in Figure 3.1(c). The free-form piping design made possible by additive manufacturing could potentially reduce the overall weight and space occupied by the networking component. More importantly, additive manufacturing process generally doesn't require any specific new tooling, which could save a lot of time in the process of prototype production and validation. If the prototype reveals that the design is pending further revisions, a new design can be generated timely, while traditional approach requires longer time to process, which could also be a waste of material and labor (Lipson and Kurman, 2013).

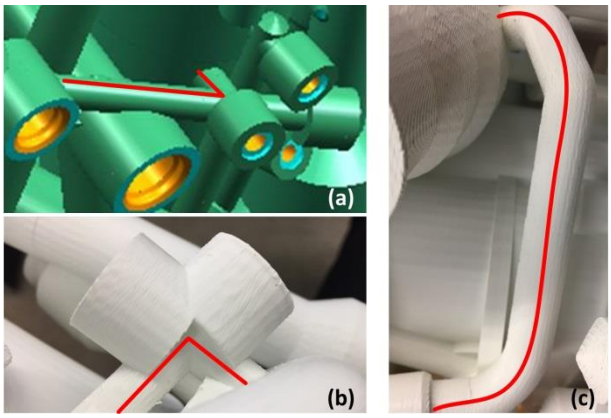
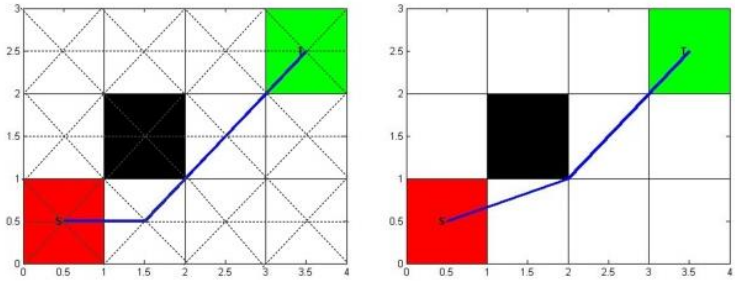


Figure 3.1. Piping design for traditional manufacturing vs. piping design for additive manufacturing: (a) A piping design blueprint for traditional manufacturing. (b) A piping design prototype for traditional manufacturing. (c) A piping design prototype for additive manufacturing (implemented by the proposed approach).

Mathematically, the problem of length optimization of networking components under given constraints, i.e., locations of connection elements, can be preliminarily casted into a known terrain path-finding framework that is frequently encountered in video games, GPS navigation, and robotics and path planning etc (Rabin, 2002; Sturtevant and Geisberger, 2010; Konar et al, 2013; Rakshit et al, 2013; LaValle, 2006; Cui et al, 2016). Other geometric or physical constraints, such as curvatures, pressure

drops, and etc., are successively addressed either manually or through an automated procedure. Known terrain path-finding can be generally divided into two steps (Algfoor et al, 2015): 1) discretizing the continuous routing space, and 2) searching along the graph to find the path minimizing the cost value, which represents the overall length of the path from the starting node to the target node for a non-weighted graph. In this study, without losing generality, square grids are used to discretize the terrain owing to its simplicity and popularity in various applications. A review of different terrain discretizations can be found in (Algfoor et al, 2015). One of the earliest investigations was the Dijkstra's algorithm (1959) where the cost value for the incremental search to the nearest goal is used. The well-known A* algorithm (Hart et al, 1968) made some improvements by adding heuristic cost from the current node to the target node to the evaluation. A* is widely used for solving path-finding problems as it has simple algorithmic structure and is guaranteed mathematically to find a solution. However, the shortest path in gridded graph is not equivalent to the shortest path in continuous space where polynomials along the edge of grids can be replaced by straight lines (Rabin, 2000). Figure 3.2(a) illustrates the shortest graph path obtained by A* algorithm. Since A* algorithm can only take horizontal, vertical or diagonal propagation steps (i.e., zigzag steps) towards neighbor nodes, the path heading (direction) is constrained by propagation headings. The shortest continuous path is depicted in Figure 2.2(b), where the path length is apparently shorter than that in Figure 3.2(a). A commonly adopted solution to address the optimality issue is to apply post-processing techniques to shorten the obtained grid path at the cost of an increase of computational time (Thorpe and Matthies, 1984; Botea et al, 2004). The post processing techniques usually find a shorter any-angle path compared to A*, but the improvement is limited (Zitzler, 1999) since they directly make adjustment to paths obtained by A* without knowing whether the post-processed path is optimal or not (Zhang and Li, 2007).



(a) (b)

Figure 3.2. (a) The shortest graph path. (b) The shortest continuous path.

Aiming at overcoming the limitations of A*, Field D* (Ferguson and Stentz, 2006), an advanced version of D* Lite (Koenig and Likhachev, 2002), was proposed which uses the linear interpolation of path costs of vertices to obtain any-angle paths. However, Field D* could result in paths that have unnecessary heading changes and should thus be smoothed further (Nash and Koenig, 2013). Some other approaches, such as Theta* (Daniel et al, 2010; Nash et al, 2007) and Lazy Theta* (Nash et al, 2010), were also proposed which embeds smoothing process into the A* searching to release the constraint on path searching directions (45 or 90 degrees). Theta* finds shorter paths in less time than Filed D* according to literature (Nash et al, 2007). Nevertheless, it is not optimal because the parent of a vertex has to be either a visible neighbor of the vertex or the parent of a visible neighbor, which is not always the case for true shortest paths (Daniel and Nash, 2010; Dang et al, 2015). Figure 3.3 shows the optimal path found by Theta* and the true shortest path, in which the path found by Theta* takes unnecessary detours compared to the shortest path. Similar to Theta*, Accelerated A* (Šišlák et al, 2009) is an any-angle path-finding algorithm that may find better path than Theta*. But it could include a large proportion of nodes in the Closed List, i.e., large number of node evaluations for challenging problems (Harabor and Grastien, 2013).

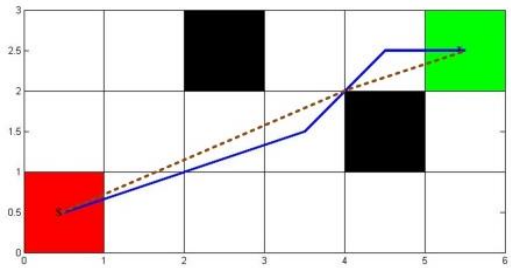


Figure 3.3. Path found by Theta* vs. true shortest path

The majority of existing path-finding algorithms are for applications such as video games and navigations where computational efficiency, i.e., finding a sub-optimal solution instantly, is the most important factor. It is worth noting that the requirement on path-finding for additive-manufacturing oriented design optimization is considerably different. In the optimization of a piping network, the optimality of the path becomes critically important, while the computational efficiency takes a second role as long as the time it takes remains to be reasonable for design iteration. A known approach of finding the shortest paths in known terrain with polygonal obstacles is A* on Visibility Graphs (Lozano-Pérez and Wesley, 1979). While A* on Visibility Graphs is guaranteed to identify the shortest path, the computational cost is extremely high because it propagates along the edges of a visibility graph, the number of which grows exponentially with the size of the gridded map (see Section 3.3.3 for computational complexity analysis). Moreover, for different starting node and target node setups in the same map, visibility graphs have to be generated repeatedly. Even though the process can be accelerated by performing visibility check dynamically or using reduced visibility graphs (Mitchell and Papadimitriou, 1991; Liu and Arimoto, 1992), A* on Visibility Graphs is still very slow and cannot be used directly in piping design as the analysis has to be carried out reiteratively due to multiple connections in each iteration of the system. Path-planning algorithms such as continuous Dijkstra and its variants (Mitchell et al, 1987; Hershberger and Suri, 1999) as well as the recent Anya (Harabor and. Grastien, 2013) also attempt to find the shortest path but have not yet been thoroughly evaluated. Generally speaking, to balance between optimality and efficiency, one would need to design an any-angle path-finding algorithm that can find a path shorter than that of the basic A* on Grids and at the same time be faster than A* on Visibility Graphs.

In this chapter, a new Focal Any-Angle A* approach is proposed to demonstrate the possibility and capability of additive manufacturing in the area of piping network design. A pruned visibility graph based on Candidate Vertices, a term we use to represent the vertices that the optimal path may pass

through, is developed first, followed by a ray-casting-based visibility check technique. The new approach preserves the optimality advantage of visibility graph-based methods, and reduces the number of evaluations needed for path-finding compared to both gird-based (A* on Grids, Theta*) and visibility graph-based (A* on Visibility Graphs) methods. The performance of the new algorithm, hereafter referred to as Focal Any-Angle A* (FA-A*) in this paper, is compared to A*, Theta* and A* on Visibility Graphs in terms of path length, nodes evaluated as well as computational time. Eventually, the developed algorithm is applied to piping network design problems where optimal paths are obtained under reasonable computational time. While the performance of some piping networks may not be merely related to the length of the pipes, we also proposed one possible approach to handle other physical or geometrical constraints. The overall design methodology has been successfully validated through a networking component prototype fabricated by Stereolithography (SLA).

3.2 Algorithm components

In this section, we outline the components and techniques that will be used in the proposed approach, namely, Candidate Vertices acquisition, visibility check, and post smoothing. Without loss of generality, all examples given are in 2D. 3D cases can be implemented similarly.

3.2.1 Candidate Vertices acquisition

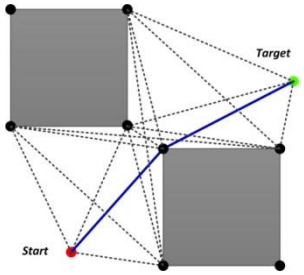


Figure 3.4. Visibility Graph of two nodes and two obstacles, and the shortest path.

The Visibility Graph of a gridded map contains the starting node, the target node and all the vertices of obstacle blocks (Lozano-Pérez and Wesley, 1979). The reason why we need such a visibility graph is because the true shortest paths have heading changes only at the vertices of the blocks. But the vertices that the true shortest path may pass by are comprised of only a small subset of all vertices (Figure 3.4).

Those are the vertices that we really want to keep track of. Here we introduce a new method to identify the possible vertices that the optimal path may pass through. We call this subset of vertices the Candidate Vertices, denoted as **CV**. All the Candidate Vertices along with starting node and target node and their edges constitute the pruned visibility graph. The method is explained in detail below which includes two steps, a one-time pre-processing, and a dynamic **CV** generation process.

3.2.1.1 Pre-processing

For a problem with *n* obstacles, an *n* by *n* symmetric proximity matrix **D** is used to record pairwise distances. For example, the proximity value between the *i*-th obstacle and the *j*-th obstacle is specified by the distance between the centers of them, denoted by *x* and *y*, if obstacles are unit squares,

$$\mathbf{D}(i, j) = |i, j| \tag{3.1}$$

The obstacles that have proximity values smaller than $\sqrt{2}$ will be clustered if diagonal move between obstacles is allowed. If not, we cluster the pairs with proximity values less or equal to $\sqrt{2}$ (Figure 3.5). Here single linkage method is applied for clustering obstacles (Jain and Richard, 1988).

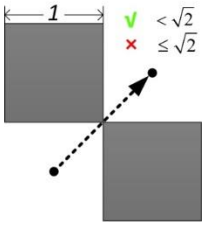


Figure 3.5. Diagonal move between obstacles.

Next, for each cluster, quick hull algorithm is adopted to obtain the vertices on the convex hull of each cluster. It is possible that when non-convex clusters are tightly placed against each other like mortise and tenon, the optimal path may go through the vertices that are not on the convex hull of the clusters. Thus, it is recommended to consider all the vertices on the corners of a cluster when the proportion of obstacles is large or when we are dealing with maze routing problems. However, for obstacle-avoidance problems such as piping design, vertices on convex hull are adequate and thus will be adopted in the first three case studies and the piping design problem in this paper (Section 3.4.1 - 3.4.5). In the fourth numerical test (Section 3.4.4), vertices on the corners will be considered. After eliminating

the vertices on boundaries, the remaining vertices are the vertices that will be further used to construct the **CV**, and we call this set of vertices $\mathbf{V}_1$. An example path-finding problem after pre-processing can be found in Figure 3.6(a) where 36 vertices ($\mathbf{V}_1$) are in the pool for **CV**.

The pre-processing only needs to be performed once for one gridded map regardless the positions of starting node and target node. For piping design problems, usually multiple pipelines are to be designed for a single 3D gridded graph (Section 3.4.5), the pre-processing, however, only need to be carried out once. On the other hand, full visibility graphs need to be constructed for different starting node and target node setups. Another merit of this pre-processing technique is that, no matter how small the grid size is or how many obstacles are used to capture the silhouette of one blocking feature; it can be always regarded as one cluster instead of a large number of obstacles, which enhances the computational performance of the algorithm along the line when maps have more and more details.

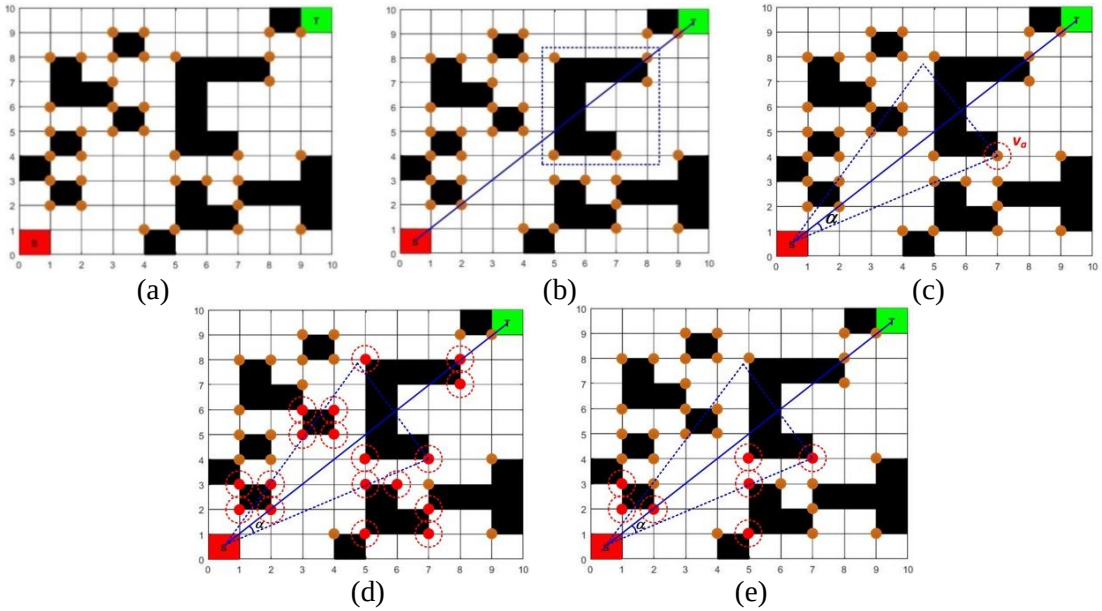


Figure .36. An example of finding Candidate Vertices. (a) Vertices in V_1, (b) Blocking cluster and V_2 (inside the box), (c) Vertex corresponding to the largest angle, (d) Vertices in V_3, (e) Candidate Vertices (marked with dotted circles).

3.2.1.2 CV generation

After pre-processing, the Candidate Vertices set **CV** will be dynamically generated throughout the searching process. In each step, the clusters that block the straight line between the current node and the target node are considered first. In other words, we first investigate the vertices on convex hulls of the

clusters that block the straight line between the current node and the target node.

Figure 3.6 illustrates the steps of generating **CV**. As shown in Figure 3.6(b), the dotted box indicates the blocking cluster and associated vertices. We name this set of vertices $\mathbf{V}_2$ which is a subset of $\mathbf{V}_1$. The visibility is checked to determine whether any cluster blocks the straight line between two nodes. The visibility check technique adopted will be introduced in Section 3.2.2. Next, we find the vertex on either side of the straight line in $\mathbf{V}_2$ that constructs the largest angle (α) with the straight line between the current node and the target node. α essentially quantifies the extent of cluster blocking the connection between the current node and the target node. The vertex corresponding to α is denoted as $\boldsymbol{v}_a$. Subsequently, an isosceles triangle, or a cone in 3D cases, can be formed, as shown in Figure 3.6(c). Implicitly, clusters of the same shape that stay close to the current node will have higher possibility in terms of containing $\boldsymbol{v}_a$. Assume the current node is represented by $\boldsymbol{p}$, the target node is $\boldsymbol{q}$ and a vertex is $\boldsymbol{v}$. We can determine the relative position of $\boldsymbol{v}$ to the straight line between the current node $\boldsymbol{p}$ and the target node $\boldsymbol{q}$. The straight line defined by $\boldsymbol{p}$ and $\boldsymbol{q}$ is,

$$Ax + By + C = 0 \tag{3.2}$$

where $A = p_y - q_y$, $B = q_x - p_y$, and $C = p_x \cdot q_y - q_x \cdot p_y$.

Substituting the position of the vertex into the above equation, we have,

$$D = Av_x + Bv_y + C \tag{3.3}$$

The position of the vertex $\boldsymbol{v}$ with respect to the straight line can be determined by comparing D to 0. Then the angle α can be calculated accordingly, i.e.,

$$\alpha = \arccos \frac{\overrightarrow{\boldsymbol{pq}} \cdot \overrightarrow{\boldsymbol{pv}}}{|\boldsymbol{pq}||\boldsymbol{pv}|} \tag{3.4}$$

We now have a triangle, or a cone in 3D cases, that is defined by the current node, and $\boldsymbol{v}_a$. The vertices in $\mathbf{V}_1$ that belong to the clusters inside or tangential to this triangle are narrowed down as the vertices set $\mathbf{V}_3$ (Figure 3.6(d)). Here we use the algorithm proposed in (Hormann and Agathos, 2001) to examine whether a point is inside a polygon. Finally, we check if the vertices in $\mathbf{V}_3$ are visible from the current

node; all the visible ones are the so-called Candidate Vertices set **CV**. The **CV**, along with the current node, the target node and their edges constitute the pruned version of visibility graphs (Figure 3.6(e)). Then the vertices in **CV** are used in A* algorithm as neighboring nodes instead of the nodes adjacent to the current node. Instead of propagating to all adjacent nodes as in A* on Grids, or propagating to all visible vertices as in A* on Visibility Graphs, the search now propagates to only the vertices in **CV**, the size of which is significantly smaller than that of all adjacent nodes and all visible vertices. A flowchart is drawn to articulate the process of generating **CV** iteratively as a part of path-finding algorithm in Figure 3.7. We are able to reduce the computational cost of A* on Visibility Graphs by checking visibility of the vertices in $\mathbf{V}_3$ instead of all the vertices and only propagating to **CV** whereas the key idea of A* on Visibility Graphs has been preserved.

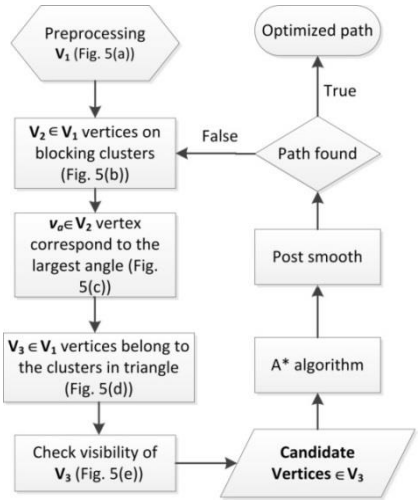


Figure 3.7. Flowchart of finding the Candidate Vertices as part of path-finding.

3.2.2 Visibility check

In Theta*, a visibility check technique called line-of-sight is adopted which is derived from (Bresenham, 1965). The idea of line-of-sight check is straightforward. It checks certain grids between two nodes based on their relative position. If the examined grid is an obstacle, the two nodes are non-visible to each other. Line-of-sight checks can be performed efficiently with only integer operations on square grids. However, such technique is ad-hoc when adapted to other discretizations or different searching rules because the grids to examine are different case by case. Thus, in this research, we

propose a ray-casting algorithm to check for visibility between two nodes based on the algorithm proposed in (Amanatides and Woo, 1987). The computational complexity of the two approaches is similar ($O(n)$), but the approach used in this paper is more flexible and can accommodate piping network design applications systematically.

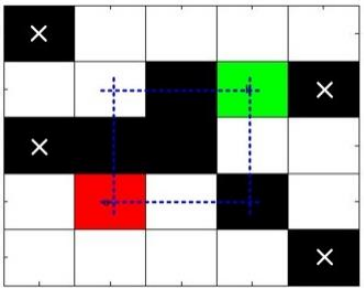


Figure 3.8. The obstacles in between starting node and target node.

The steps of checking the visibility between two nodes with the existence of obstacles are outlined below:

Step 1. Consider the obstacles occurring between the nodes. Observe Figure 3.8. When an obstacle occurs between the nodes, the *x*, *y* values of the obstacle in Cartesians system cannot exceed the maximum or minimum *x*, *y* values of the two nodes. The obstacles marked with a cross in Figure 3.8 should be eliminated. If two nodes have the same *x* value or *y* value, then the visibility can be determined directly by examining the grids in between.

Step 2. Check if the obstacles block the ray between nodes. We check the obstacles that are bounded by the nodes (Figure 3.8) one after another to see if any of them intersect with the line segment between the two nodes. Consider Figure 3.9 as an example, in which $\boldsymbol{p}$ and $\boldsymbol{q}$ are two nodes, ***Obs*** is a unit obstacle where the vertices with the smallest and largest *x*, *y* values are $\boldsymbol{v}_{min}$ and $\boldsymbol{v}_{max}$, respectively. The line segment connecting $\boldsymbol{p}$ and $\boldsymbol{q}$ goes through the obstacle (intersect), if the following conditions are satisfied

$$\frac{tx_{\min}}{\left|(\overrightarrow{\boldsymbol{pq}})_x\right|} < \frac{ty_{\max}}{\left|(\overrightarrow{\boldsymbol{pq}})_y\right|} \tag{3.5}$$

$$\frac{ty_{\min}}{\left|(\overrightarrow{\boldsymbol{pq}})_y\right|} < \frac{tx_{\max}}{\left|(\overrightarrow{\boldsymbol{pq}})_x\right|} \tag{3.6}$$

In Equations (3.5) and (3.6), $\left|(\overrightarrow{\boldsymbol{pq}})_x\right|$ and $\left|(\overrightarrow{\boldsymbol{pq}})_y\right|$ are projections of $|\boldsymbol{pq}|$ on *x*-axis and *y*-axis, $tx_{\min}$, $ty_{\min}$ are projections of $|\boldsymbol{pv}_{\min}|$ on *x*-axis and *y*-axis, and $tx_{\max}$, $ty_{\max}$ are projections of $|\boldsymbol{pv}_{\max}|$ on *x*-axis and *y*-axis, as shown in Figure 3.9. If we change '<' to '≤' in Equations (3.5) and (3.6), merely passing through a vertex will be considered as intersection.

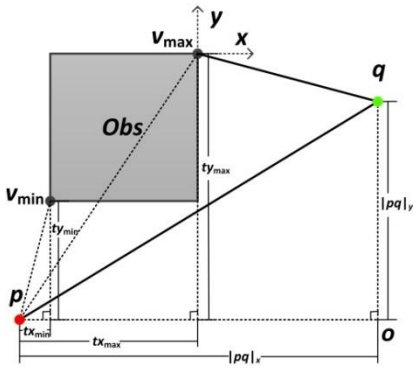


Figure 3.9. Visibility check example.

Once an obstacle is calculated to have intersected with the line segment between the nodes, we can terminate the examination and conclude that the two nodes are non-visible to each other. For more details, interested readers may refer to (Amanatides and Woo, 1987).

3.2.3 Post smoothing

The post smoothing technique adopted is proposed in (Daniel et al, 2010), which updates the parent of a vertex when using A*. The pseudo-code of the post smoothing technique is given below. In the pseudo-code, **visibilityCheck**(n_1, n_2)=1 means node n_1 and n_2 are visible to each other using the visibility check technique outlined in Section 3.2.2. The post smoothing technique aims at reducing unnecessary heading changes.

```
postSmoothing(n)
    if visibilityCheck(n, parent(parent(n))) = 1
        if g(parent(parent(n))) + c(parent(parent(n)), n)<g(n)
            parent(n)= parent(parent(n))
  g(n)= g(parent(parent(n)))+ c(parent(parent(n)), n)
            open.Update(n, parent(n), g(n), h(n))
        end if
    end if
```

end

3.3 New approach formulation

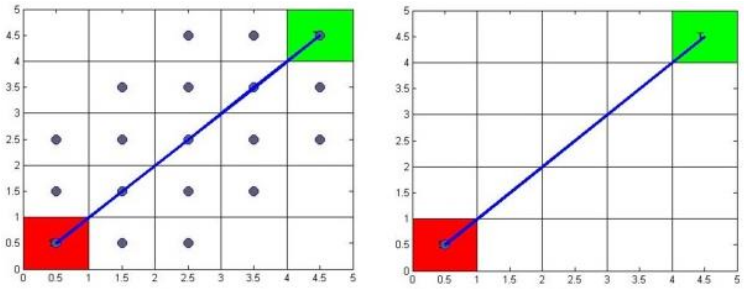
3.3.1 FA-A* algorithm

We propose an optimal path-finding approach based on the techniques outlined in Section 3.2. The A* algorithm that uses the techniques in Section 2 is referred to as FA-A* (Focal Any-Angle A*). The pseudo code for FA-A* is provided below.

```
Main()
    open = ∅
    g(n_start):=0
    parent(n_start) = n_start
    open.Insert(n_start, parent(n_start), g(n_start), h(n_start))
    n_current = n_start
    while n_current exists
        propagate = CV of n_current
        if n_target ∈ propagate
            open.Insert(n_target, n_current, g(n_curren)+c(n_curren, n_target),0)
            return "path found"
        end if
        open.Refresh (propagate, n_current)
        open.Close(n_current)
        n_current = the node in open with the smallest g+h
    end while
end

open.Refresh(propagate, n_current)
    for every n' ∈ propagate
        if n' ∈ open
            if g(n_current)+c(n_current, n')<g(n') value in open
                open.Update(n', n_current, g(n_current)+c(n_current, n'), h(n'))
            end if
        else
            open.Insert(n', n_current, g(n')_new, h(n'))
        end if
        postSmoothing(n')
    end for
end
```

In the pseudo-code, *n* means node. Therefore n_{start}, n_{target}, $n_{current}$, are the starting node, the target node, and current node, respectively. The notation $g(n)$ represents the shortest path length from the starting node to node *n* found so far. $c(n_1, n_2)$ is the cost travel from n_1 to n_2, i.e., the Euclidean distance between n_1 and n_2. And $h(n)$ is the heuristic cost from node *n* to target node. In this research, the heuristic cost is the Euclidean distance between two nodes.



(a) (b)

Figure 3.10. Node evaluations comparison between A* on grids and FA-A*

The key difference between FA-A* and A* is that FA-A* only propagates to Candidate Vertices (**CV** introduced in Section 3.2.1) of the current node instead of all the neighboring nodes (please see the grayed line in the pseudo-code). Consider a simple case where there is no obstacle. A* or Theta* need to evaluate 17 nodes along the search until they find the target node (Figure 10(a)), whereas FA-A* could find the target node directly from starting node because the target node is one of the Candidate Vertices of the starting node (Figure 3.10(b)). Note that in the pseudo-code, we only maintain an *open* list, whereas in A* and Theta* a *closed* list co-exists with the *open* list. Such change is facilitated by the *open.Close* operation which prevents the present current node from being either the current node or appearing in the **CV** again in the future using an indicator. This change may help to enhance the performance of FA-A* in terms of efficiency.

3.3.2 Optimality analysis

The selection process of Candidate Vertices proposed could be regarded as a greedy approach to reduce the number of visibility checks and function evaluations compared to the methods that use complete visibility graphs. The algorithm is inspired by the scenarios when two nodes are connected with

rubber band as the optimal rubber band connections always have the shortest distance possible and have heading changes mostly at or in between the vertices that belong to the clusters that play the biggest roles in blocking two nodes. The role played by the obstacles in blocking two nodes is an abstract notion which is quantified in our research as α. Then vertex $\boldsymbol{v_a}$ correspond to the largest outreaches of the blocking clusters on each side. By specifying $\mathbf{V}_3$ and **CV**, our proposed approach (FA-A*) indeed can pinpoint the possible vertices that the optimal path may pass through. Moreover, it preserves the optimality advantages of using visibility graphs. In our case studies (Section 4), the proposed method always finds the shortest paths for all 21 path graphs. The optimality of the algorithm has yet to be proved mathematically and will be addressed in our future works.

3.3.3 Computational complexity analysis

This section provides the complexity analysis of FA-A*. The operations of FA-A* and their worst-case complexity are outlined as follows.

1) Prepossessing: $O(|\boldsymbol{Obs}| \log |\boldsymbol{Obs}|)$ as it is the complexity for quick hull algorithm. $|\boldsymbol{Obs}|$ is the number of obstacles.
2) Check which clusters block the straight line between starting node and current node: $O(n)$. $O(n)$ is the complexity for visibility check where n represents the number of obstacles in between the two nodes.
3) Calculate α : $O(|\mathbf{V}_2|)$. $|\mathbf{V}_2|$ is the size of $\mathbf{V}_2$.
4) Check if a vertex is inside the triangle: $O(3|\mathbf{V}_1|)$. $O(3)$ is the complexity to check if a point is inside of a triangle.
5) Check if vertices in $\mathbf{V}_3$ are visible from current node: $O(n|\mathbf{V}_3|)$.

In summary, the computational complexity for FA-A* is

$$O(\text{FA-A*}) = O(|\boldsymbol{Obs}| \log |\boldsymbol{Obs}|) + O(Expan_{\text{FA-A*}}) \times O(n + |\mathbf{V}_2| + 3|\mathbf{V}_1| + n|\mathbf{V}_3|) \tag{3.7}$$

where $Expan_{\text{FA-A*}}$ is the number of expansion in FA-A*. This can be simplified to

$$O(\text{FA-A*}) = O(Expan_{\text{FA-A*}} \times (n + |\mathbf{V}_2| + 3|\mathbf{V}_1| + n|\mathbf{V}_3|)) \tag{3.8}$$

The first term in Equation (3.7) is omitted because that the second term has the dominant effect on complexity as the scale of the problem increases.

In comparison, the computational complexity of A* is $O(8\times Expan_{\text{A*}})$, the complexity of Theta* is $O((8+n)\times Expan_{\theta*})$, and the complexity of A* on Visibility Graphs is $O(n|\mathbf{V}|\times Expan_{\text{A*onV}})$ where $|\mathbf{V}|$ is the number of vertices. For grid-based approaches (A* on Girds, Theta*), the number of expansion is directly related to the grid size of the graph. Take Figure 10 for example, $Expan_{\text{A*}}$ and $Expan_{\theta*}$ increase linearly as the grid size decreases while $Expan_{\text{FA-A*}}$ and $Expan_{\text{FA-A*}}$ remain the same. As will be discussed later and demonstrated in Section 3.4.3, the number of expansion of graph-based methods (A* on Visibility Graphs, FA-A*) is more related to the number and detail level of obstacle clusters. Usually for piping network design problems where the grid size is small for better design details and the obstacles are mostly appear in chunks rather than scattered, the number of expansion for each algorithm has the following empirical relations,

$$Expan_{\text{A*}} > Expan_{\text{FA-A*}} \tag{3.9}$$

$$Expan_{\theta*} > Expan_{\text{FA-A*}} \tag{3.10}$$

$$Expan_{\text{A*onV}} > Expan_{\text{FA-A*}} \tag{3.11}$$

Since **V** is a set of all the vertices, if the following inequality holds,

$$n|\mathbf{V}| > n+|\mathbf{V}_2|+3|\mathbf{V}_1|+n|\mathbf{V}_3| \tag{3.12}$$

FA-A* will be more efficient than A* on Visibility Graphs in terms of computational complexity, given Equations (3.11) and (3.12). Since the sizes of $\mathbf{V}_1$, $\mathbf{V}_2$ and $\mathbf{V}_3$ are proportional to the number of clusters, FA-A* is efficient in solving problems with large clusters of obstacles rather than discrete small obstacles with the same proportion of obstacles, which is validated in in Section 3.4.3. As a result, FA-A* has advantages over methods that propagate to immediate neighbors like A* and Theta* computationally when handling maps that have small number of clusters of obstacles. Meanwhile, FA-A* is always more efficient than A* on Visibility Graphs. All of the above could make FA-A* ideal in solving either

obstacle avoidance problems such as piping design or maze routing problems.

3.3.4 Data structure

In this research, we maintain a concise data structure where two main matrices are used. The first matrix, denoted as $\mathbf{V}_{all}$, is a $|\mathbf{V}|$ by four (five for 3D cases) matrix, where $|\mathbf{V}|$ is the number of all vertices. Each row of $\mathbf{V}_{all}$ corresponds to one vertex where the first two columns (three columns in 3D case) are the position of the vertex in the coordinate system, the second to last column is the obstacle index to which the vertex belongs, and the last column is the cluster index to which the obstacle belongs. Similarly, the second matrix, denoted as $\mathbf{V}_{convex}$, is a $|\mathbf{V}_1|$ by four (five for 3D cases) matrix where each row corresponds to one vertex in $\mathbf{V}_1$. $\mathbf{V}_{all}$ is used to determine $\mathbf{V}_1$ in pre-processing, and $\mathbf{V}_{convex}$ is formed accordingly. After using $\mathbf{V}_{convex}$ to determine $\mathbf{V}_2$, we use $\mathbf{V}_{all}$ to check whether a cluster is inside or tangential to the triangle. Finally, $\mathbf{V}_{convex}$ can be used to determine $\mathbf{V}_3$ and $\mathbf{CV}$.

3.4 Validation and case demonstration

In this section, we first evaluate the performance of the proposed approach (FA-A*) by comparing with with A* on Grids (A* on G), Theta*, and A* on Visibility Graphs (A* on V) under 21 maps that belong to four representative test scenarios as have been used in (Sturtevant and Geisberger, 2010; Daniel et al, 2010; Dang et al, 2015; Yap et al, 2011). FA-A* is able to find the optimal path for all 21 cases with tractable efficiency. Then FA-A* is adopted to facilitate an example freeform piping network design problem. Specifically, the proposed approach is employed to design a network with 12 pipelines, and the design achieved is compared to that of A* on Grids. The metrics employed are the length of the path (*L*), the number of node evaluations (*No.*), and the computational time (*T*). The heuristic value *h* used in the case studies is the Euclidean distance from the current node to the target node. The computational time recorded is the average over five runs. All algorithms are implemented in MATLAB and executed on a 2.40 GHz 2 processors (Xeon E5620) computer.

3.4.1 Placing starting and target nodes in the corners

In this sub-section, 10 maps with different complexities are used as the starting node and the target

node are placed respectively at the bottom left corner and the top right corner.

We start off by comparing the algorithms on 50*50 gridded graphs. The obstacles are randomly generated with occupations of 5%, 10%, 20%, 30% and 50% of the map, respectively. As shown in the test results plotted in Figure 11 and listed in Table 3.1, FA-A* always finds the optimal/shortest path. As demonstrated, FA-A* outperforms the A* and Theta* in all obstacle scenarios in terms of path length and number of node evaluations. Moreover, it is always faster than A* on Visibility Graphs, and faster than A* on Grids and Theta* when the proportion of obstacles is 5%. Figure 3.12 shows the path/shortest path versus the runtime which indicates how the algorithms balance between optimality and efficiency. Figure 3.12(a), (c), (e) and (i) show that FA-A* stays in the left bottom corner indicating the best overall performance and a good compromise of the grid-based technique and the visibility graph-based technique. For the case with 5% obstacles, as can be seen in Figure 3.12(a), FA-A* prevails in all metrics.

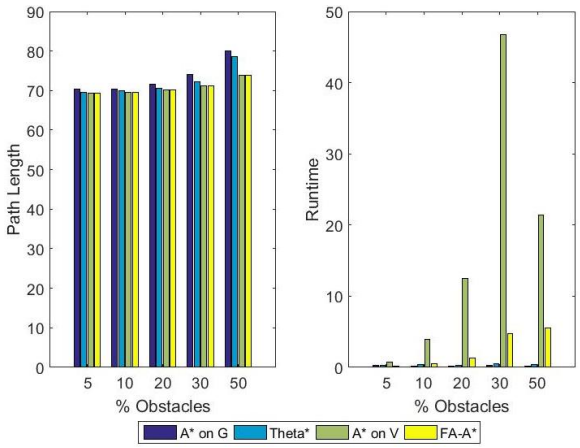
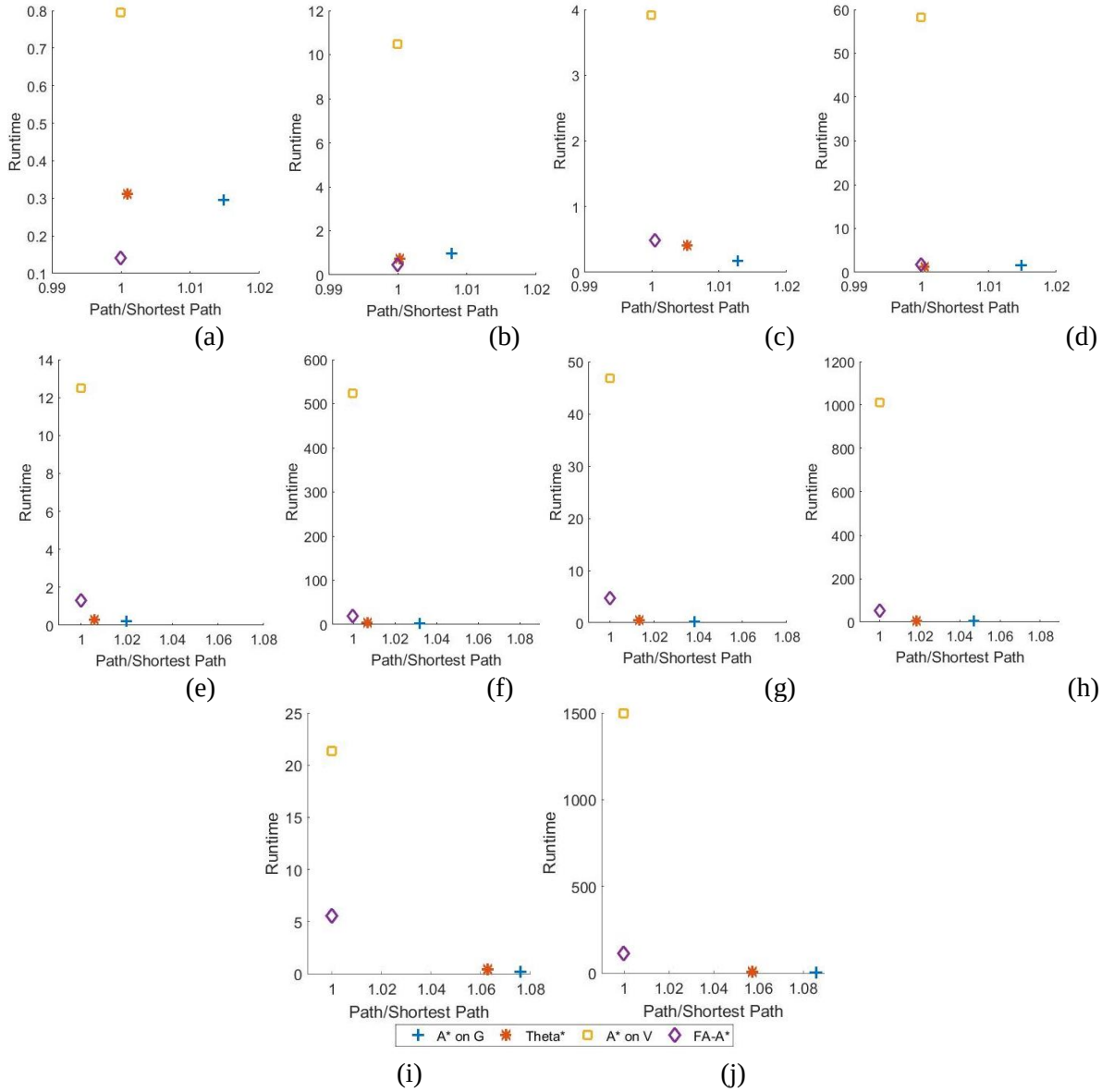


Figure 3.11. Random 50*50 maps with different proportion of obstacles

Table 3.1 Performance on 50*50 graphs

Map Info	**Algorithm**	***L***	***No.***	***T*** (sec)
50*50 5%	A* on G	70.4680	487	0.2964
	Theta*	69.5051	406	0.3120
	A*on V true shortest	69.4394	392	0.7956
	FA-A*	69.4394	31	0.1404
50*50 10%	A* on G	70.4680	416	0.1716
	Theta*	69.9440	505	0.4056
	A*on V true shortest	69.5763	705	3.9156
	FA-A*	69.5763	122	0.4992
50*50	A* on G	71.6396	483	0.2028

20%	Theta*	70.6594	429	0.3120
	A*on V true shortest	70.2469	828	12.4997
	FA-A*	70.2469	207	1.3232
50*50 30%	A* on G	73.9828	668	0.2808
	Theta*	72.1964	551	0.4680
	A*on V true shortest	71.2623	839	46.7961
	FA-A*	71.2623	353	4.7736
50*50 50%	A* on G	79.9828	589	0.2184
	Theta*	78.5568	519	0.4056
	A*on V true shortest	73.9302	358	21.3549
	FA-A*	73.9302	240	5.5632



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j)

Figure 3.12. Runtime versus path length. (a) 50*50: 5%, (b) 100*100: 5%, (c) 50*50: 10%, (d) 100*100: 10%,(e) 50*50: 20%, (f) 100*100: 20%,(g) 50*50: 30%, (h) 100*100: 30%,(i) 50*50: 50%,(j) 100*100: 50%.

We then proceed to 100*100 graphs with the same obstacles fractions as in the 50*50 scenario. The results are reported in Table 3.2 and visualized in Figure 3.13. A demonstration of paths found by each algorithm and the comparison can be found in Figure 3.14 where the evaluated nodes are filled with dots. Our observation from this set of experiments is consistent with 50*50 cases. When there are a small number of obstacles, FA-A* not only finds the best path, but also excels in computational speed. When the obstacle fraction increases, the paths found by FA-A* are shorter than those of A* and Theta* significantly. Such observation is also supported by Figure 3.12. The paths found by FA-A* with and without post smoothing are compared in Figure 3.15. The only difference is zoomed in while the rest of two paths coincide with each other. Even without post smoothing, FA-A* without post smoothing still finds better path than A* or Theta* as it is only 0.007428% longer than the true shortest path.

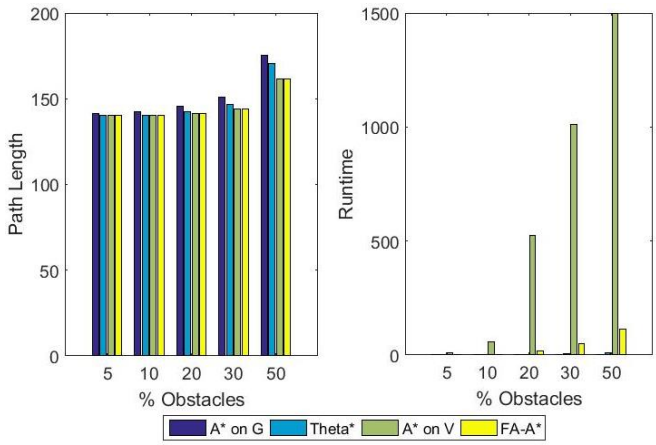
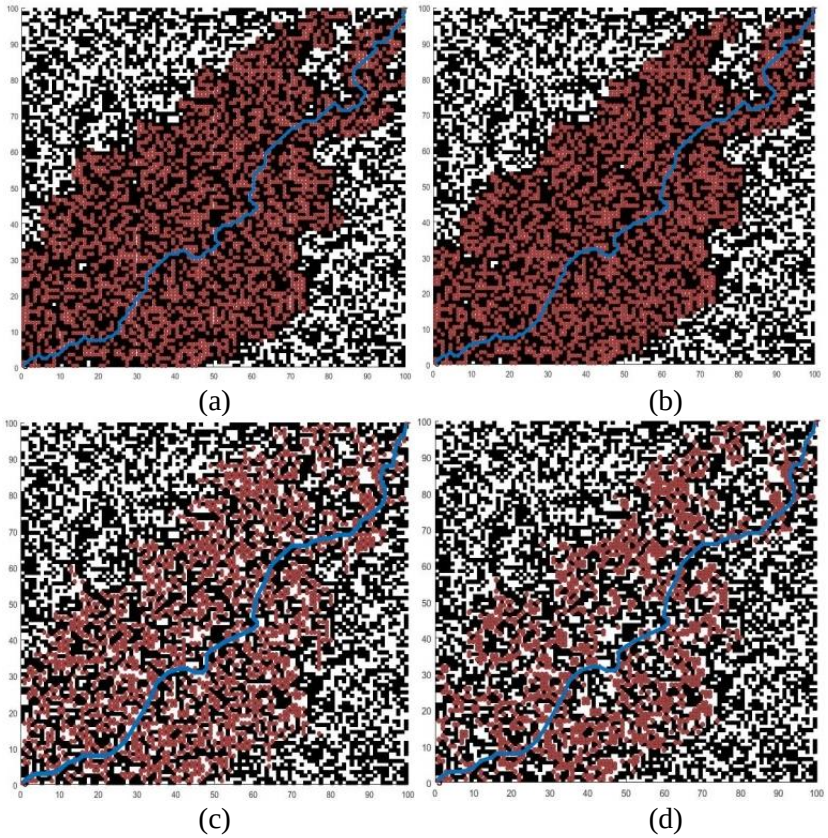


Figure 3.13. Random 100*100 maps with different proportion of obstacles

Table 3.2 Performance on 100*100 graphs

Map Info	**Algorithm**	***L***	***No.***	***T*** (sec)
100*100 5%	A* on G	141.1787	950	0.9828
	Theta*	140.1210	723	0.7332
	A*on V true shortest	140.0883	1345	10.4793
	FA-A*	140.0883	62	0.4680
100*100 10%	A* on G	142.3503	1364	1.5756
	Theta*	140.3394	958	1.2948
	A*on V true shortest	140.2633	2365	58.3068

	FA-A*	140.2633	193	1.7964
100*100 20%	A* on G	145.8650	2344	3.1668
	Theta*	142.3683	1653	3.4632
	A*on V true shortest	141.3687	2625	524.1037
	FA-A*	141.3687	662	19.3401
100*100 30%	A* on G	150.7939	2658	3.4788
	Theta*	146.7182	2029	5.4756
	A*on V true shortest	144.0496	2280	1011.5
	FA-A*	144.0496	1305	57.0032
100*100 50%	A* on G	175.3797	2889	3.6284
	Theta*	170.8014	2706	9.1977
	A*on V true shortest	161.5234	2193	1498.0
	FA-A*	161.5234	1491	112.6314



(a) (b)

(c) (d)

Figure 3.14. Paths found and nodes evaluated by each algorithm (100*100, 50%). (a) A* on G,(b) Theta*,(c) A* on V,(d) FA-A*.

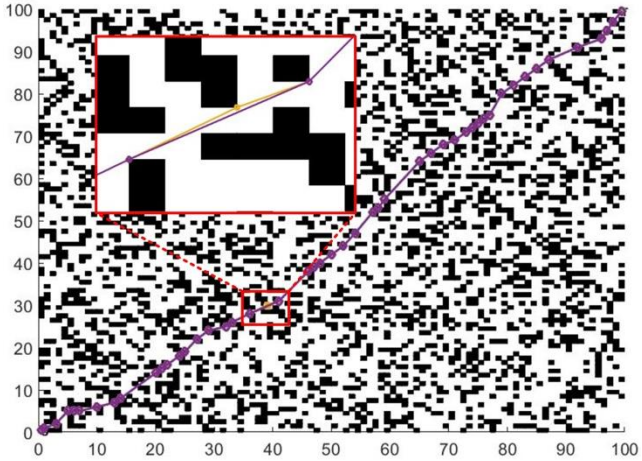


Figure 3.15. Paths comparison of FA-A* with and without post smoothing (100*100, 50%)

3.4.2 Placing starting node in middle

Next we apply the algorithms to 300*300 maps with various proportions of obstacles. The starting node is placed in the middle of the map (150, 150), and the target node is randomly placed. The results are demonstrated in Figure 3.16 and Table 3.3. Because it took A* on Visibility Graphs too long to find the optimal paths for last two cases, the results are omitted. The performance comparison of the other three algorithms is our focus here. FA-A* always finds better paths than A* on Grids and Theta*. Moreover, FA-A* shows the best performance in all three metrics when the fraction of obstacles are 5% and 10%.

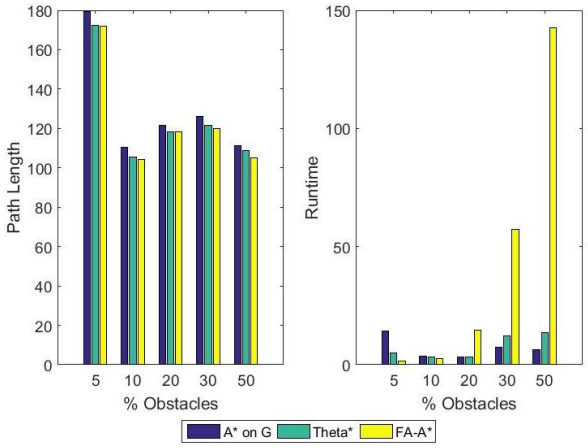
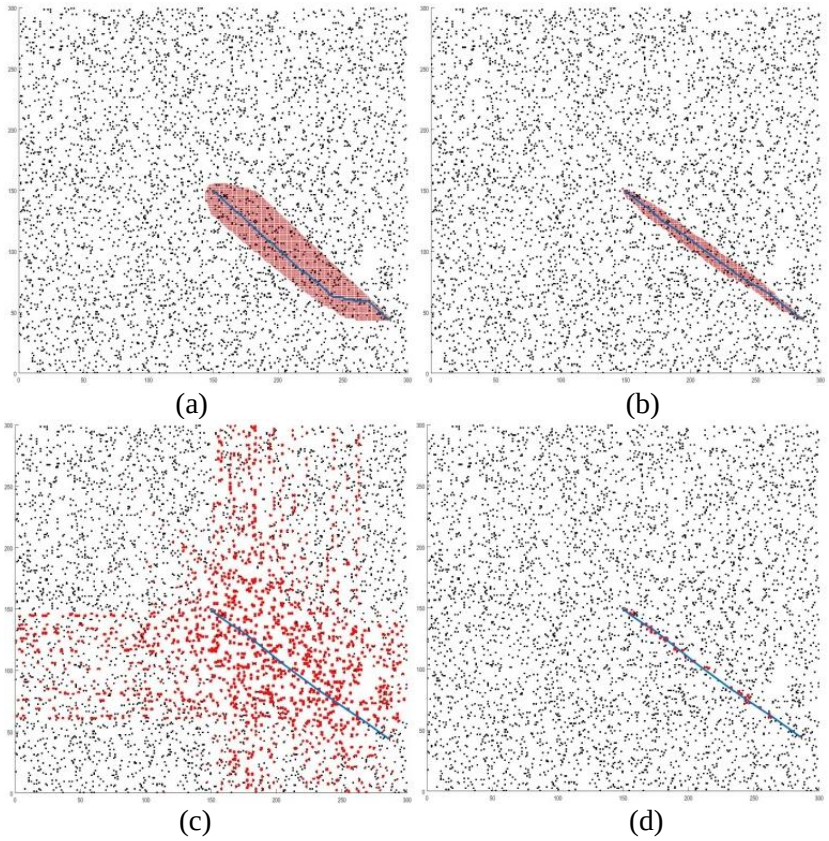


Figure 3.16. Random 300*300 maps with different proportion of obstacles



(a) (b) (c) (d)

Figure 3.17. Paths found and nodes evaluated by each algorithm (300*300, 5%). (a) A* on G, (b) Theta*, (c) A* on V, (d) FA-A*.

A comparison of paths is provided in Figure 3.17, where A* on Grids evaluates a large number of nodes (4911) around the path found and Theta* managed to reduce the number of node evaluations compared to A* on Grids (1619). Meanwhile, A* on Visibility Graphs evaluates 4589 nodes regardless the position of the target as shown in Figure 3.17, and FA-A* only evaluates 137 nodes in the searching process which is significantly smaller than the other three algorithms.

Table 3.3 Performance on 300*300 graphs

Map Info	**Algorithm**	***L***	***No.***	***T*** (sec)
300*300 5% Target (286, 45)	A* on G	179.4924	4911	14.1165
	Theta*	172.3074	1619	4.9140
	A*on V true shortest	171.9521	4589	171.9521
	FA-A*	171.9521	137	1.7160
300*300 10%	A* on G	110.2843	2090	3.7284
	Theta*	105.3346	967	3.1512
	A*on V	104.3420	4470	1826.2

Target (170, 48)	true shortest			
	FA-A*	104.3420	168	2.8704
300*300 20% Target (230, 236)	A* on G	121.4802	1359	3.1824
	Theta*	118.4848	757	3.3384
	A*on V true shortest	118.1886	2991	8894.4
	FA-A*	118.1886	389	14.6573
300*300 30% Target (50, 212)	A* on G	126.2670	2211	7.5192
	Theta*	121.5660	1391	12.2617
	A*on V true shortest	N/A	N/A	N/A
	FA-A*	119.9381	759	57.1432
300*300 50% Target (75, 111)	A* on G	111.2965	1327	6.2712
	Theta*	108.6661	1217	13.6813
	A*on V true shortest	N/A	N/A	N/A
	FA-A*	105.0499	706	142.4913

The results in Section 3.4.1 and 3.4.2 are consistent in the sense that FA-A* outperforms three other algorithms in all categories when the number of obstacles is small. Even though the efficiency of FA-A* deteriorates as the number of obstacles increases, it still tops A* on Visibility Graphs. Based on our complexity analysis, the main factor that affects the efficiency of FA-A* is the number of clusters rather than the number of obstacles. It will be validated in Section 3.4.3. Naturally, when the obstacles are randomly generated, the number of clusters is proportional to the number of obstacles.

3.4.3 Different number of clusters

Here we investigate the impact of the number of clusters. In freeform piping design, the obstacles are clustered together to capture the details of the blocking features. Thus we use 50*50 maps of 30% of obstacles with varying numbers of clusters to emulate piping design scenarios. The position, size and shape of the cluster are all randomly generated. The results are reported in Table 3.4. As observed, FA-A* always finds the shortest path and the time used decreases as the number of clusters decreases, while the runtime of A* on G and Theta* do not show much of an improvement. A comparison of the four algorithms with varying numbers of clusters is illustrated in Figure 3.18. When the number of clusters is 35, as shown in Figure 3.19, FA-A* dominates in all categories. Evidently, it is superior to methods that

propagate to immediate neighbors for problems with small number of clusters. As in piping design, obstacles are usually clustered to represent components in the system, FA-A* is therefore more practical.

Table 3.4 Performance on 50*50 graphs

Map Info	**Algorithm**	***L***	***No.***	***T*** (sec)
50*50 30% 316 clusters	A* on G	75.9828	793	0.4056
	Theta*	73.7824	633	0.5928
	A*on V true shortest	72.2600	741	32.3234
	FA-A*	72.2600	372	5.1948
50*50 30% 75 clusters	A* on G	75.1543	687	0.2964
	Theta*	72.0079	401	0.2808
	A*on V true shortest	71.2361	254	2.9484
	FA-A*	71.2631	102	1.2168
50*50 30% 57 clusters	A* on G	74.5685	594	0.2496
	Theta*	71.6930	453	0.3900
	A*on V true shortest	71.2190	178	2.4024
	FA-A*	71.2190	84	0.8892
50*50 30% 35 clusters	A* on G	74.5685	946	0.5928
	Theta*	71.5985	743	0.8112
	A*on V true shortest	71.1266	124	0.9360
	FA-A*	71.1266	36	0.2964
50*50 30% 1 cluster	A* on G	85.6985	1403	0.7800
	Theta*	80.0536	762	0.5304
	A*on V true shortest	79.8618	4	0.0356
	FA-A*	79.8618	4	0.0624

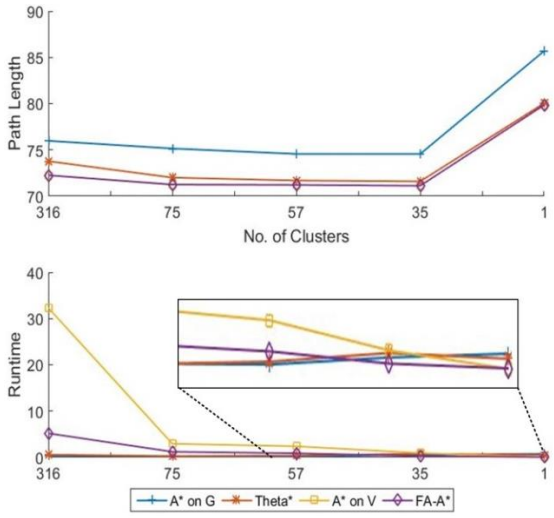
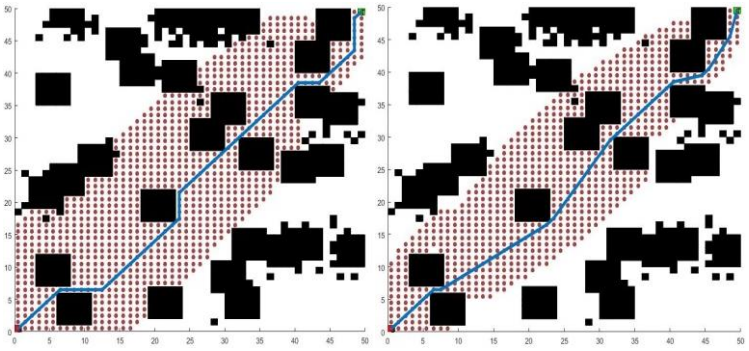


Figure 3.18. Performance comparison (50*50)


(a) (b)

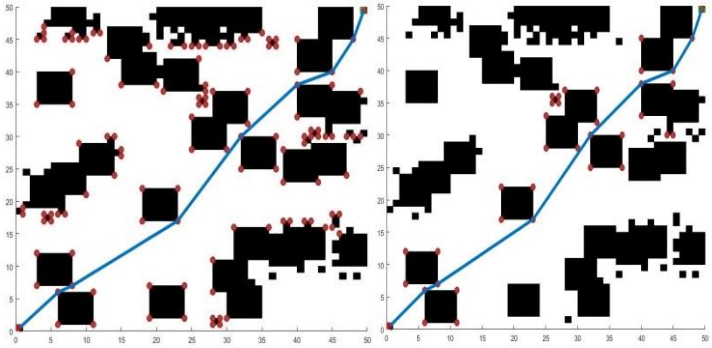

(c) (d)

Figure 3.19. Paths found and nodes evaluated by each algorithm (300*300, 35 clusters). (a) A* on G,(b) Theta*,(c) A* on V,(d) FA-A*.

3.4.4 Maze routing

We further validate the findings reported in the preceding sub-sections by using one of the benchmark maze routing problems. It is a 511*511 map with 2.79% obstacles and 17 clusters. The starting node and the target node are chosen as (15, 466) and (466, 15). The results are shown in Table 3.5. FA-A* finds the shortest path among the four algorithms. Besides, as shown in Figure 3.20, A* on Grids and Theta* are 'trapped' in the maze and evaluate a large number of nodes throughout (149,689 and 147,896) whereas FA-A* only evaluates 104. Hence, FA-A* is 98.71% faster than A* on G and 99.23% faster than Theta*. Additionally, consistent with all previous case studies (Figure 3.21), FA-A* has the least node evaluations, which helps it to gain an advantage in computational time.

Table 3.5 Performance on a 511*511 maze

Map Info	**Algorithm**	***L***	***No.***	***T*** (sec)
511*511	A* on G	1436.8	149689	2877.5
2.79%	Theta*	1393.9	147896	4901.4

17 clusters	A*on V true shortest	1389.9	120	36.9722
	FA-A*	1389.9	104	37.1282

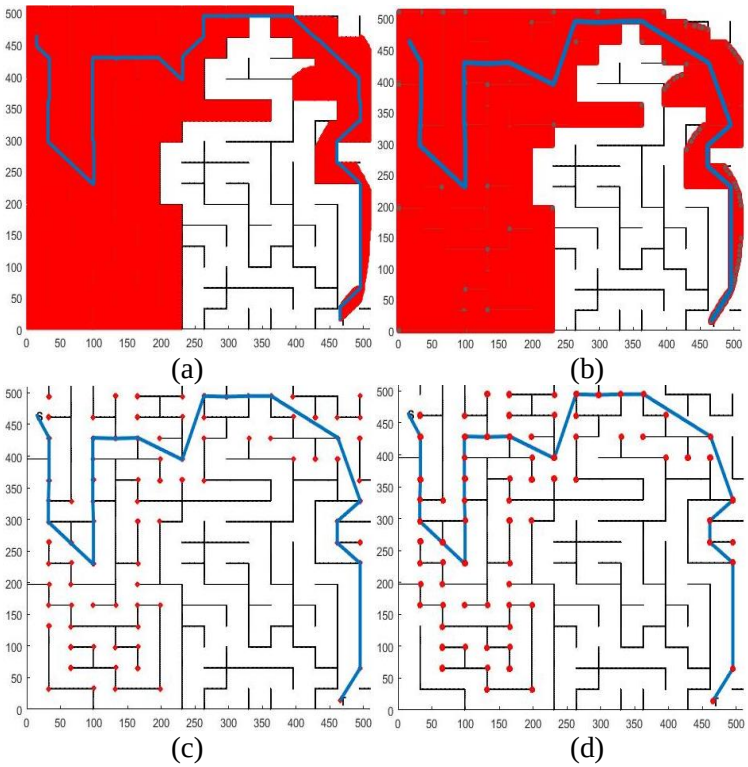


Figure 3.20. Path found for 511*511 maze routing. (a) A* on G,(b) Theta*,(c) A* on V,(d) FA-A*.

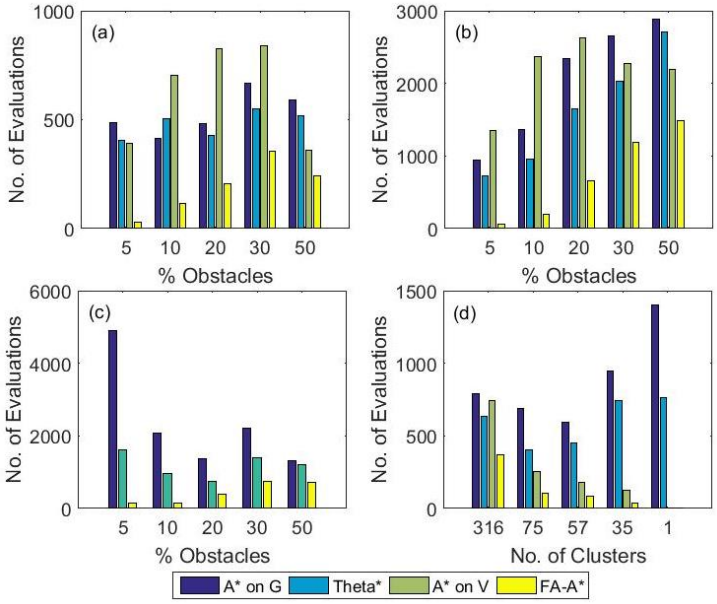
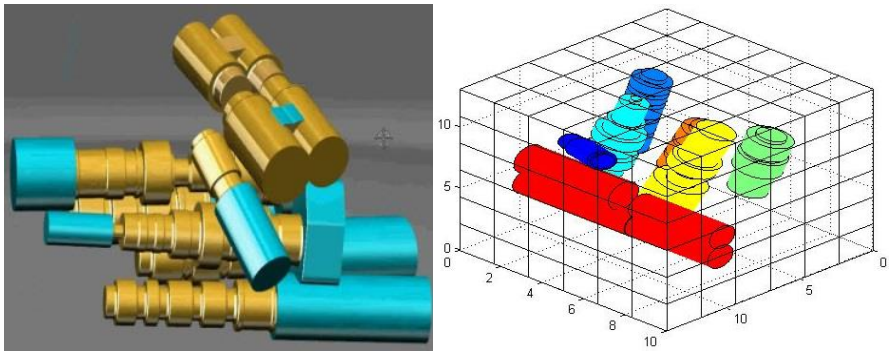


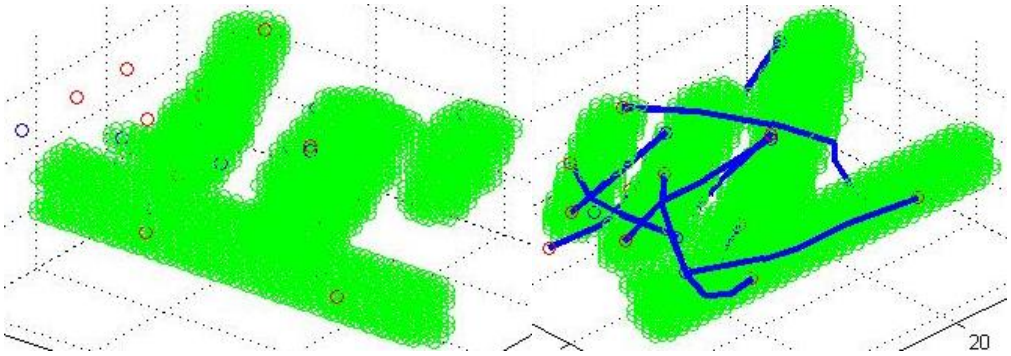
Figure 3.21. Number of evaluations comparison. (a) 50*50 (b) 100*100 (c) 300*300 (d) 50*50 with different no. of clusters

3.4.5 Freeform piping network design demonstration

Hydraulic systems are frequently encountered in industrial applications, consisting of pumps, valves, reservoirs, and various actuators etc. that are connected with piping networks. In a typical design of hydraulic systems, the locations of pumps, valves, actuators as well as certain inlets and outlets are decided first, followed by the design of piping networks. The rapid advancement of additive manufacturing technology yields a new means of fabricating certain hydraulic systems, which entertains advantages including less constraint in assemblage and enhanced design freedom with the possibility of incorporating freeform piping networks. As the locations of other components are decided, the piping network essentially determines the total volume, weight, and cost of the final design. Therefore, the optimality of the piping length is emphasized. Meanwhile, the computational cost has to be reasonable to accommodate design iterations. The proposed FA-A* algorithm is specifically tailored for such applications, i.e., obtaining the optimal paths of piping network design with increased efficiency. While the optimality and efficiency are validated in the preceding sub-sections, in what follows we demonstrate a representative freeform piping network design problem (Figure 3.22(a)). Here the locations of the components as well as the locations of inlets and outlets are the starting and target nodes.



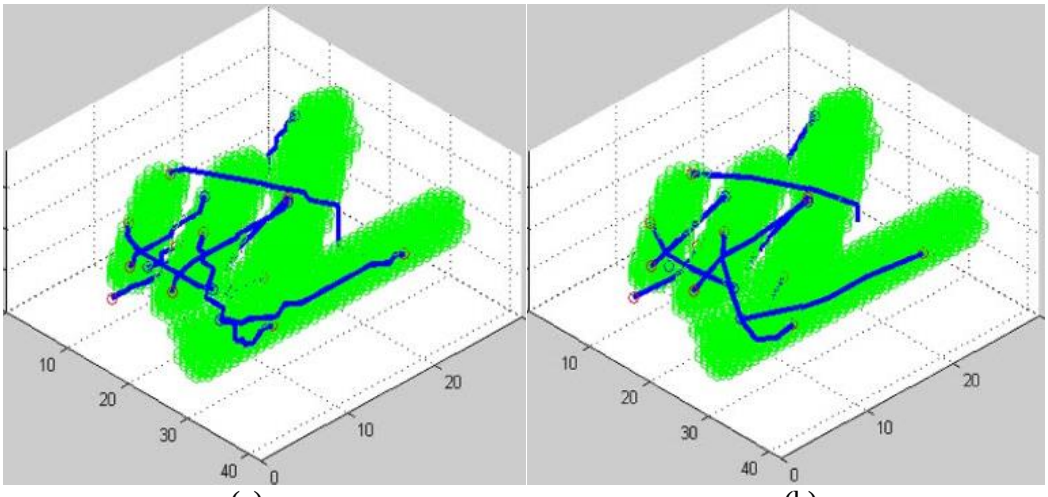
(a) (b)



(c) (d)

Figure 3.22. Piping design example. (a) CAD model of pre-specified components, (b) Geometry model, (c) Gridding, (d) Optimal paths.

The piping network design procedure in this study can be summarized in four steps. First, given an actual CAD model of the pre-specified components (Figure 3.22(a)), the geometry is outlined in a 3-dimensional space (Figure 3.22(b)). Next, the geometry is gridded into points (Figure 3.22(c)) to facilitate piping design. The gird size can be adjusted to accommodate different accuracy requirements. FA-A* is then implemented to find the optimal paths, i.e., freeform network of piping, between pre-specified pairwise starting and target nodes (Figure 3.22(d)). Finally, the optimal paths obtained are cast into the CAD model to generate the final design. In this demonstration, the pre-specified starting and target nodes are marked in Figure 3.22(c). A total number of 12 pipelines are to be designed, i.e., 12 paths are to be identified. The size of the gridded map is 160*120*40.



(a) (b)

Figure 3.23. Piping design comparison. (a) A* on G, (b) FA-A*.

Figure 3.23 shows the comparison of results by A* on Grids and FA-A*. FA-A* not only yields better piping design, but also excels in efficiency. As illustrated in Table 3.6, the total length and computational time of using FA-A*are 738.1940 and 223, respectively, both of which are the best among the four algorithms. Compared to A*, an 11.53% length reduction is obtained using FA-A*, which is a significant improvement in terms of weight reduction. Moreover, FA-A* features a 99.645% reduction in the number of evaluation compared to A* on Grids and a 97.7311% reduction in computational time compared to A* on Visibility Graphs.

Table 3.6 Performance on a 240*120*40 piping design problem

Map Info	**Algorithm**	***L***	***No.***	***T*** (sec)
160*120*40 27% obstacle 7 clusters	A* on G	834.4152	215210	270
	Theta*	817.6547	214930	253
	A*on V true shortest	738.1940	2817	9833
	FA-A*	738.1940	764	223
	FA-A*: % of improvement compared to the second best	9.718%	72.879%	11.858%

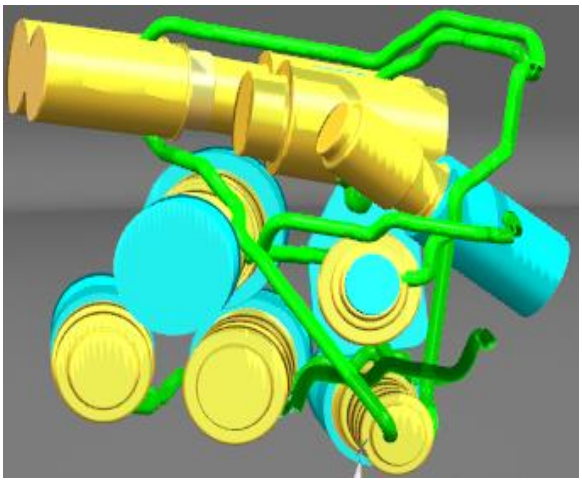


Figure 3.24. Optimal piping design. Inlet and outlet locations of the system are pre-specified at the top-right and bottom-right in the figure.

The piping network design processes of A* on Grids and FA-A* are animated as shown in https://youtu.be/kCkHjn2Ga8I and https://youtu.be/0rg1u78BLHw, respectively. As can be observed, the paths identified by FA-A* are smoother and may indeed avoid drastic pressure drop (less turns). As shown in the video, the pair of starting node and target node that has the shortest straight-line distance is connected first. The optimal paths found are subsequently regarded as obstacles for the succeeding paths. The design sequence is indeed subject to change for different networking components where certain pipelines are more critical than others. While our focus is primarily placed on the optimality of individual paths, the overall sequence should be determined based on domain expertise and empirical experience. As seen in the videos, while A* shows noticeable detours in its path-finding process, FA-A* creates the shortest/optimal connections between each pair of nodes and yield better designs. After casting the design in 3D map into the CAD model, Figure 24 shows the final piping network design identified by FA-A*.

Although the shortest piping design can generally avoid drastically pressure drops, the performance of certain piping networks may not be merely related to the length or pressure drop. Here we also briefly outlines (Figure 3.25) a perturbation-based optimization approach to handle other physical or geometrical constraints as means to refine the piping design. The FA-A* design will serve as the initial design where the points that define the FA-A* path are regarded as control points. Then by adding, removing, or moving one or a set of control points, a new path can be generated and evaluated based on its quantifications related to desired performance, e.g., length, curvature or turns, hydraulic resonant and etc. Such practice is iterated and whether to accept the perturbation can be determined by the stochastic optimization approach employed, such as evolutionary optimization, particle swarm optimization, and simulated annealing. The overall methodology can accommodate various networking component designs when certain performance aspects are required and additive manufacturing is the designated fabrication approach.

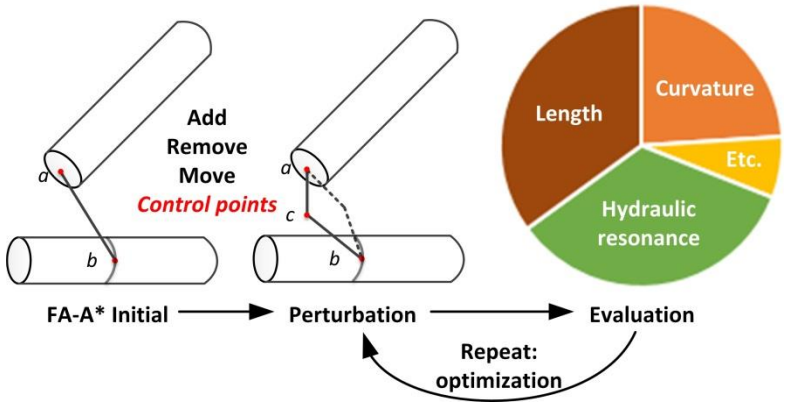


Figure 3.25. Path refinement through perturbation-based optimization

One prototype of the freeform piping design assisted by FA-A* has been successively printed using Stereolithography (SLA). As the overall design methodology does not cater to specific additive manufacturing techniques, eventually, depending on the material, final products can be fabricated by Powder Bed Fusion (PBF) process including additive manufacturing techniques such as Selective Laser Sintering (SLS) or Electron Beam Melting (EBM). As illustrated in Figure 3.26, when there is no obstacle in between, a straight pipeline is designed; otherwise, optimal curves are generated.

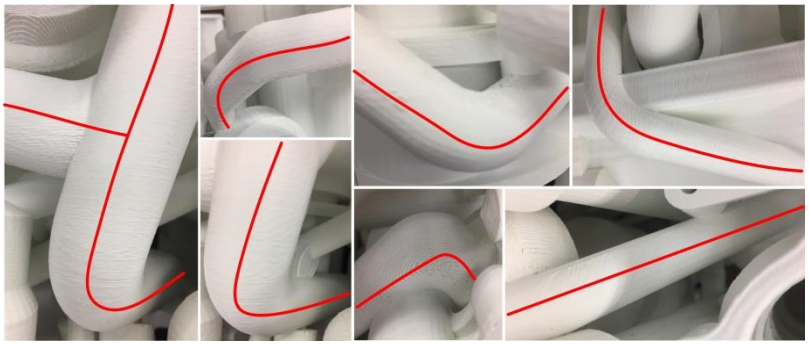


Figure 3.26. Stereolithography (SLA) prototype of a freeform piping network designed by FA-A*

3.5. Conclusion

In this research, we develop a focal any-angle A* algorithm based on visibility graphs (FA-A*) towards piping network to demonstrate the capability of additive manufacturing to such industrail design. FA-A* performs the path-finding with focuses on possible paths to the target, and prunes the full visibility graph accordingly. As demonstrated in the case studies, due to the use of visibility graphs, the

algorithm is capable of finding the shortest path. Meanwhile, because of the reduction of the visibility graphs, FA-A* enhances the computational performance of exact algorithms. Moreover, FA-A* has the least node evaluations for all the test cases. For problems that have more complicated path costs, FA-A* would gain significant advantages in terms of computational time. Overall, FA-A* not only preserves the optimality of visibility graph-based methods but also can keep up with grid-based methods computationally. Case studies validate that the newly developed approach finds better paths than A* and Theta*, and yields better computational efficiency than A* on Visibility Graphs. It is further demonstrated that the algorithm can find the shortest piping path with tractable computational cost, which provides a viable design tool for the additive manufacturing of freeform piping network.

Chapter 4. A Reinforcement Learning Hyper-Heuristic in Multi-Objective Single Point Search with Application to Structural Fault Identification

Multi-objective optimizations are frequently encountered in engineering practices. The solution techniques and parametric selections however are usually problem-specific. In this study we formulate a reinforcement learning hyper-heuristic scheme, and propose four low-level heuristics which can work coherently with the single point search algorithm MOSA/R (Multi-Objective Simulated Annealing Algorithm based on Re-seed) towards multi-objective optimization problems of various properties. Making use of the domination amount, crowding distance and hypervolume calculations, the proposed hyper-heuristic scheme can meet various optimization requirements adaptively. The approach developed not only exhibits better and robust performance compared to AMOSA, NSGA-II and MOEA/D when applied to benchmark test cases, but also shows promising results when applied to a generic structural fault identification problem. The outcome of this research can be extended to many design and manufacturing optimization applications.

4.1 Introduction

Many engineering optimization problems involve multiple types of goals, thus naturally present themselves as multi-objective problems. For example, the rapid advancement of sensing and measurement technologies has made it possible to realize structural fault identification in near real-time. Fault parameters in a structure are generally identified through matching measurements with model predictions in the parametric space. Since multiple measurements are usually involved, the identification can be cast into a multi-objective optimization problem.

Multi-objective optimization algorithms have been practically applied to a variety of problems, ranging from production scheduling (Wang et al, 2014; Lu et al, 2016), structural design (Kaveh and Laknejadi, 2013), performance improvement (Szollos et al, 2009), to structural fault pattern recognition (Cao et al, 2018a; 2018b) etc. The solution techniques, nevertheless, are often devised and evaluated for specific problem domains, which not only require in-depth understanding of the problem domain

involved but are also difficult to be exercised to different instances. Even for the same type of problems, the formulation may need to be changed as more knowledge and insights are gained. The hyper-heuristic concept was therefore suggested (Cowling et al, 2000), aiming at producing general-purpose approaches. The terminology implies that a high-level scheme to select heuristic operators is incorporated as the detailed algorithms are being executed (Burke et al, 2009) given a particular problem and a number of low-level heuristics. Instead of finding good solutions, hyper-heuristic is more interested in adaptively finding good solution methods. Since its emergence, the subject has gained significant interests, and a number of studies of hyper-heuristic have been performed for multi-objective problems. Burke et al (2007) and Sabar et al (2011) proposed hyper-heuristic approaches to address timetabling and scheduling problems. Gomez and Terashima-Marin (2010), de Armas et al (2011) and Bai et al (2012) extended the hyper-heuristic method to handle packing and space allocating problems. Raad et al (2010) and McClymont and Keedwell (2011) used hyper-heuristics to water resource and distribution problems. Wang and Li (2010) and Vazquez-Rodrigues and Petrovic (2013) also applied hyper-heuristic framework to multi-objective benchmark problems such as DTLZ and WFG. More recently, Guizzo et al (2015) applied hyper-heuristic based multi-objective evolutionary algorithms to solve search-based software engineering problems. Hitomi and Selva (2015; 2016) investigated the effect of credit definition and aggregation strategies on multi-objective hyper-heuristics and used it to solve satellite optimization problems. Interested readers may refer to (Burke et al, 2013; Maashi et al, 2015) for more discussions about hyper-heuristic techniques and applications.

Typically, a hyper-heuristic framework involves: (1) a high-level selection strategy to iteratively select among low-level heuristics based on the performance; (2) a predefined repository of low-level heuristics; and (3) applying the selected heuristics into optimization and evaluating their performance. The selection mechanism in hyper-heuristics, which essentially ensures the objectivity, specifies the heuristic to apply in a given point of optimization without using any domain information. With this in mind, online learning hyper-heuristics usually take advantage of the concept of reinforcement learning for selection (Kaelbling et al, 1996; Ozcan et al, 2012), as it aims to iteratively solve the heuristics selection

task by weight adaptation through interactions with the search domain. The low-level heuristics correspond to a set of exploration rules, and each carries a utility value. The values are updated at each step based on the success of the chosen heuristic. An improving move is rewarded, while a worsening move is punished. The low-level heuristics can be embedded in single point search techniques, which are highly suited for these tasks because only one neighbor is analyzed for a choice decision (Nareyek, 2003). In a single point search-based hyper-heuristic framework, e.g., a simulated annealing (Kirkpatrick et al, 1983) based hyper-heuristic, an initial candidate solution goes through a set of successive stages repeatedly until termination.

The goal of this research is to advance the state-of-the-art in Multi-Objective Simulated Annealing (MOSA) by incorporating hyper-heuristic systematically to improve both the generality and performance. We develop a reinforcement learning hyper-heuristic inspired by probability matching (Goldberg, 1990), which consists of a selection strategy and a credit assignment strategy. As discovered in previous investigations (Cao et al, 2016 and 2017), the solution quality/diversity as well as the robustness of the algorithm can be enhanced with re-seed schemes. The re-seed schemes, on the other hand, need to be tailored to fit specific problem formulation. Here in this research the re-seed schemes are treated as the low-level heuristics, empowering the algorithm to cover various scenarios. The performance and generality of the proposed approach are first demonstrated over benchmark testing cases DTLZ (Deb et al, 2002) and UF (Zhang et al, 2008) in comparison with popular multi-objective algorithms, namely, NSGA-II (Deb et al, 2002), AMOSA (Bandyopadhyay et al, 2008) and MOEA/D (Zhang and Li, 2007). The enhanced approach is then applied to structural fault identification, a highly promising application of MOSA, to examine the practical implementation. The main contributions of this paper are 1) to introduce a new reinforcement learning hyper-heuristic framework based on MOSA with re-seed; 2) to devise a new credit assignment strategy in high-level selection for heuristic performance evaluation; and 3) to provide insights on benchmark case studies and application.

4.2. Algorithmic foundation

4.2.1. Multi-objective optimization (MOO)

Intuitively, Multi-Objective Optimization (MOO) could be facilitated by forming an alternative problem with a single, composite objective function using a weighted sum approach. Single objective optimization techniques are then applied to this composite function to obtain a single optimal solution. However, the weighted sum methods have difficulties in selecting proper weight factors especially when there is no articulated *a priori* preference among objectives. Indeed, *a posteriori* preference articulation is usually preferred, because it allows a greater degree of separation between the algorithm and the decision-making process which also enables the testing process to be conducted independently of the application (Giagkiozos et al, 2015). Furthermore, instead of a single optimum produced by weighted sum methods, MOO can yield a set of alternative solutions explicitly exhibiting the tradeoff between different objectives.

The most well-known MOO methods are probably the Pareto-based ones that define optimality in a wider sense that no other solutions in the search space are superior to Pareto optimal solutions when all objectives are considered (Zitzler, 1999). A general MOO problem of *n* objectives in the minimization sense is represented as:

$$\text{Minimize } \mathbf{y} = \boldsymbol{f}(\mathbf{x}) = (f_1(\mathbf{x}),..., f_n(\mathbf{x})) \tag{4.1}$$

where $\mathbf{x} = (x_1, x_2,..., x_k) \in \mathbf{X}$ and $\mathbf{y} = (y_1, y_2,..., y_n) \in \mathbf{Y}$. $\mathbf{x}$ is the decision vector of *k* decision variables, and $\mathbf{y}$ is the objective vector. $\mathbf{X}$ denotes the decision space while $\mathbf{Y}$ is called the objective space. When two sets of decision vectors are compared, the concept of dominance is used. Assuming **a** and **b** are decision vectors, the concept of Pareto optimality can be defined as follows: **a** is said to dominate **b** if:

$$\forall i = \{1, 2,..., n\} : f_i(\mathbf{a}) \leq f_i(\mathbf{b}) \tag{4.2}$$

and

$$\exists\, j = \{1, 2,..., n\} : f_j(\mathbf{a}) < f_j(\mathbf{b}) \tag{4.3}$$

Refer to Table 4.1. Any objective function vector which is neither dominated by any other objective function vector of a set of Pareto-optimal solutions nor dominating any of them is called non-dominated with respect to that Pareto-optimal set (Goldberg, 1989; Zitzler, 1999). The solution that corresponds to the objective function vector is a member of Pareto-optimal set. Usually $\prec$ is used to denote domination relationship between two decision vectors (Table 4.1).

Table 4.1 Domination relations

Relation	Symbol	Interpretation in objective space
a dominates **b**	$\mathbf{a} \prec \mathbf{b}$	**a** is not worse than **b** in all objectives and better in at least one
b dominates **a**	$\mathbf{b} \prec \mathbf{a}$	**b** is not worse than **a** in all objectives and better in at least one
Non- dominant to each other	$\mathbf{b} \cong \mathbf{a}$	**a** is worse than **b** in some objectives but better in some other objectives

4.2.2. Multi-Objective Simulated Annealing (MOSA)

Simulated annealing (Kirpatrick et al, 1983) is a heuristic technique drawing an analogy from physics annealing process. It was originally designed for solving single objective optimization problem, and was then extended to multi-objective context. Engrand, who is among the very first to embed the concept of Pareto optimality with simulated annealing, proposed to maintain an external population archiving all non-dominated solutions during the solution procedure (Engrand, 1998). Subsequently, several Multi-Objective Simulated Annealing (MOSA) algorithms that incorporate Pareto set (Nam and Park, 2000; Suman, 2004) have been developed. The acceptance criteria in these algorithms are all derived from the differential between the new and current solutions. However, in the presence of Pareto set, merely comparing the new solution to the current solution appears to be vague. Subsequently, there have been a few techniques proposed that use Pareto domination based acceptance criterion in MOSA (Smith, 2006; Bandyopadhyay et al, 2008; Cao et al, 2016), the merit of which is that the domination status of the point is considered with respect to not only the current solution but also the archive of non-dominated solutions found so far. It has been widely demonstrated that simulated annealing algorithms are capable of finding multiple Pareto-optimal solutions in a single run.

4.2.3. Reinforcement learning hyper-heuristic

The reinforcement learning hyper-heuristic strategy proposed in this paper can be divided into two parts, namely, heuristic selection and credit assignment. The goal is to design online strategies that are capable of autonomously selecting between different heuristics based on their credits (Burke et al, 2013).

Credit assignment rewards the heuristics online based on certain criterion, and the credits are thereafter fed to the heuristic selection strategy. It is similar to the reward assignment in reinforcement learning where the agent receives a numerical reward based on the success of an action's outcome. In this study, we develop a new credit assignment strategy based on hypervolume (Zitzler and Thiele, 1999) increments and the number of solutions newly generated to calculate the credit $c_{i,t}$,

$$c_{i,t} = e^{\frac{i(t)}{iter}} \cdot \left(\frac{(HV(PF_t) - HV(PF_{t-1})}{HV(PF_{true})} + \frac{|PF_t| - \left|\bigcap(PF_t,\, PF_{t-1})\right|}{|PF_t|} \right) \Big/ \left(i(t) - i(t-1) \right) \tag{4.4}$$

where *iter* denotes the total number of iterations, $i(t)$ is the number of iterations that has been performed at epoch t (i.e., the tth time heuristic selection has been conducted), PF_t is the Pareto front at t, and $HV(*)$ approximates the hypervolume of the Pareto front in percentage using Monte Carlo approach through N uniformly distributed samples within the bounded hyper-cuboid to alleviate the computational burden. Specifically,

$$HV(PF, r^*) = volume\left(\bigcup_{x \in PF} v(x, r^*)\right) \tag{4.5}$$

where r^* is the reference point which is set to be 1.1 times the upper bound of the Pareto front in the *HV* calculation following the recommendations in literature (Ishibuchi et al., 2010; Li et al., 2016). Therefore, in Equation (4.4), $HV(PF_t) \in [0, 1]$ is the hypervolume of the Pareto front at t, $(HV(PF_t) - HV(PF_{t-1}))$ is the hypervolume increment since the last time the heuristics are selected, and $HV(PF_{true})$ is the normalization term. The term $\frac{|PF_t| - \left|\bigcap(PF_t,\, PF_{t-1})\right|}{|PF_t|} \in [0, 1]$ computes the percentage of

newly generated solution in the current Pareto front. Both terms are dimensionless and they are summed together first then divided by $(i(t)-i(t-1))$ to evaluate the performance of a heuristic as reflected by the evolution of the Pareto front per iteration. Because it is easier for the optimizer to achieve improvements at early stage of optimization, we introduce the compensatory factor $e^{\frac{iter(t)}{iter}} \in [1, e]$ to progressively emphasize the credits earned as the optimization progresses.

Heuristic selection, as its name indicates, selects from the low-level heuristics at each time epoch. The concept is similar to the agent in reinforcement learning. One difficulty that a heuristic selection strategy would have to overcome is the exploration versus exploitation dilemma (EvE), indicating that the heuristic with the highest credits should be favored whilst the heuristics with low credits should also be occasionally selected because they may produce high quality results as the search progresses. Many heuristic selection strategies have been proposed in literature, including probability matching (PM), adaptive pursuit (Thierens, 2007), choice function (Cowling et al, 2000, Maashi et al, 2015), Markov chain models (McClymont and Keedwell, 2011) and multi-armed bandit algorithms (Krempser et al, 2012). In this paper, we devise a heuristic selection strategy with minimal number of parameters inspired by the idea of probability matching to specifically fit the online learning scheme. Given a finite set of heuristic $\mathbf{O}$, an heuristic $o_i \in \mathbf{O}$ is selected at time t with probability $p_{i,t}$ proportional to the heuristic's quality $q_{i,t}$, which is mainly determined by the credit $c_{i,t}$. The parameter t is independent of the algorithm, indicating how many times the heuristic selection has been conducted. The update rule is given as follows,

$$q_{i,t} = \alpha \cdot q_{i,t-1} + (1-\alpha) \cdot c_{i,t} \tag{4.6}$$

$$p_{i,t} = p_{\min} + (1 - |\mathbf{O}| \cdot p_{\min}) \frac{q_{i,t}}{\sum_{j=1}^{|\mathbf{O}|} q_{j,t}} \tag{4.7}$$

where $p_{\min} \in (0, \frac{1}{|\mathbf{O}|}]$ is the minimum selection probability to facilitate exploration and guarantee $p_{i,t} \in [0, 1]$. It is greater than 0 so the heuristics with low credits are also considered. The forgetting

factor $\alpha \in [0, 1]$ determines the importance of the credits received previously because the current solution may be the result of a decision taken in the past. In this study, $p_{\min}$ is chosen to be 0.1 and α is chosen to be 0.5. It is worth noting here again that $t-1$ in Equation (4.5) does not imply the iteration before *t* in optimization; it means the last time the hyper-heuristic is updated. And we only update the values that correspond to the chosen heuristic at $t-1$. For unselected heuristics we have $q_{i,t} = q_{i,t-1}$. After $p_{i,t}$ is determined using Equations (4.5) and (4.6), the lower level heuristic is chosen per its probability using roulette wheel selection method.

4.3. hyper-heuristic MOSA

With the hyper-heuristic rules defined, the MOSA algorithm and the joint hyper-heuristic scheme are presented in this section.

4.3.1. MOSA/R algorithm

The algorithm used in this study is referred to as MOSA/R (Multi-Objective Simulated Annealing based on Re-seed) which was developed by Cao et al (2016) and applied to configuration optimization. MOSA/R uses the concept of the amount of domination in computing the acceptance probability of a new solution. The algorithm was originally designed aiming at solving multi-modal optimization problems with strong constraints. It is able to provide feasible solutions more efficiently compared to traditional MOSAs due to the developed re-seed technique. Such improvement can be generalized with hyper-heuristic by making the re-seed step adaptively caters to various design preferences. The pseudo-code of MOSA/R is provided below.

Algorithm MOSA/R

Set *Tmax*, *Tmin*, # of iterations per temperature *iter*, cooling rate α, $k = 0$
Initialize the *Archive* (Pareto front)
Current solution = randomly chosen from *Archive*
While (*T* > *Tmin*)
    For 1 : *iter*
      Generate a *new solution* in the neighborhood of *current solution*
      If *new solution* dominates *k* (*k* >= 1) solutions in the *Archive*
          Update

Else if *new solution* dominated by *k* solutions in the *Archive*
Action
Else if *new solution* non-dominant to *Archive*
Action
End if
End for
$k = k+1$
$T = (\alpha^{k})*Tmax$
End While

Algorithm Update

Remove all *k* dominated solutions from the *Archive*
Add *new solution* to the *Archive*
Set *new solution* as *current solution*

Algorithm Action

If *new solution* and *Archive* are non-dominant to each other
Set *new solution* as *current solution*
Else
If *new solution* dominated by *current solution*
Re-seed
Else
Simulated Annealing
End If
End if

Algorithm Re-seed

new solution is dominated by *k* (*k* >= 1) solutions in the *Archive*
Select a heuristic from **low-level heuristics** based on hyper-heuristic strategy
Set *selected solution* following the selected **heuristic**

If $\frac{1}{1+\exp(-\Delta dom_{selected,new} / \max(T,1))} > \text{rand}(0,1)*$

Set *selected solution* as *current solution*
Else
Simulated Annealing
End if
***** rand(0,1) generates a random number between 0 to1

Algorithm Simulated Annealing

$\Delta dom_{avg} = \frac{\sum_{i=1}^{k} \Delta dom_{i,new}}{k}$

If $\frac{1}{1+\exp(\Delta dom_{avg} / T)} > \text{rand}(0,1)$

Set *new solution* as *current solution*
End if

Given two solutions **a** and **b**, if $\mathbf{a} \prec \mathbf{b}$ then the amount of domination is defined as,

$$\Delta dom_{\mathbf{a},\mathbf{b}} = \prod_{i=1,\, f_i(\mathbf{a}) \neq f_i(\mathbf{b})}^{M} (\left| f_i(\mathbf{a}) - f_i(\mathbf{b}) \right| / R_i) \tag{4.8}$$

where M is the number of objectives and R_i is the range of the ith objective (Bandyopadhyay et al, 2008). As indicated in the pseudo-code, the hyper-heuristic scheme comes into effect in Algorithm Re-seed. Whenever re-seed is triggered, first a low-level heuristic is selected from the repository based on the proposed reinforcement learning hyper-heuristic (Section 4.2.3), and then the current solution is altered using the selected low-level heuristic. Most simulated annealing related hyper-heuristic studies (Antunes et al, 2011; Bai et al, 2012; Burke et al, 2013;) use simulated annealing as the high-level heuristic to select from lower level heuristic repository to exploit multiple neighborhoods which can be regarded as variable neighborhood search mechanism. However, in this research, the proposed approach uses probability matching (PM) as the high-level heuristic and part of the MOSA/R as lower level heuristics which can be regarded as adaptive operator selection (Maturana et al, 2009). In the next sub-section, we propose four low-level heuristics for the hyper-heuristic MOSA/R.

4.3.2. Low-level heuristics

Hereafter we refer the MOSA/R with hyper heuristic scheme as MOSA/R-HH. The hyper-heuristic scheme intervenes in the re-seed scheme (Algorithm Re-seed) which essentially differs MOSA/R-HH (hyper-heuristic MOSA/R) from other MOSA algorithms. In this paper we propose four re-seed strategies as low-level heuristics.

(1) The solution in the *Archive* with the minimum amount of domination. The first strategy selects the solution from *Archive* that corresponds to the minimum difference of domination amount with respect to the *new solution*. For $\forall \mathbf{x} \in$ *Archive* that dominates the *new solution*,

$$\begin{aligned} \mathbf{x}_{select} &= \arg\min_{\mathbf{x}} (\Delta dom_{\mathbf{x},\mathbf{x}_{new}}) \\ &= \arg\min_{\mathbf{x}} \left(\prod_{i=1,\, f_i(\mathbf{x}) \neq f_i(\mathbf{x}_{new})}^{M} (\left| f_i(\mathbf{x}) - f_i(\mathbf{x}_{new}) \right| / R_i) \right) \end{aligned} \tag{4.9}$$

Then the selected solution is set as *current solution* with probability $\frac{1}{1+\exp(-\Delta dom_{selected,\,new} / \max(T,1))}$. The solution corresponding to the minimum difference of

domination amount is chosen to avoid premature convergence. An example is given in Figure 4.1(a), the solution selected using this strategy corresponds the one in the *Archive* that dominates the current solution the least.

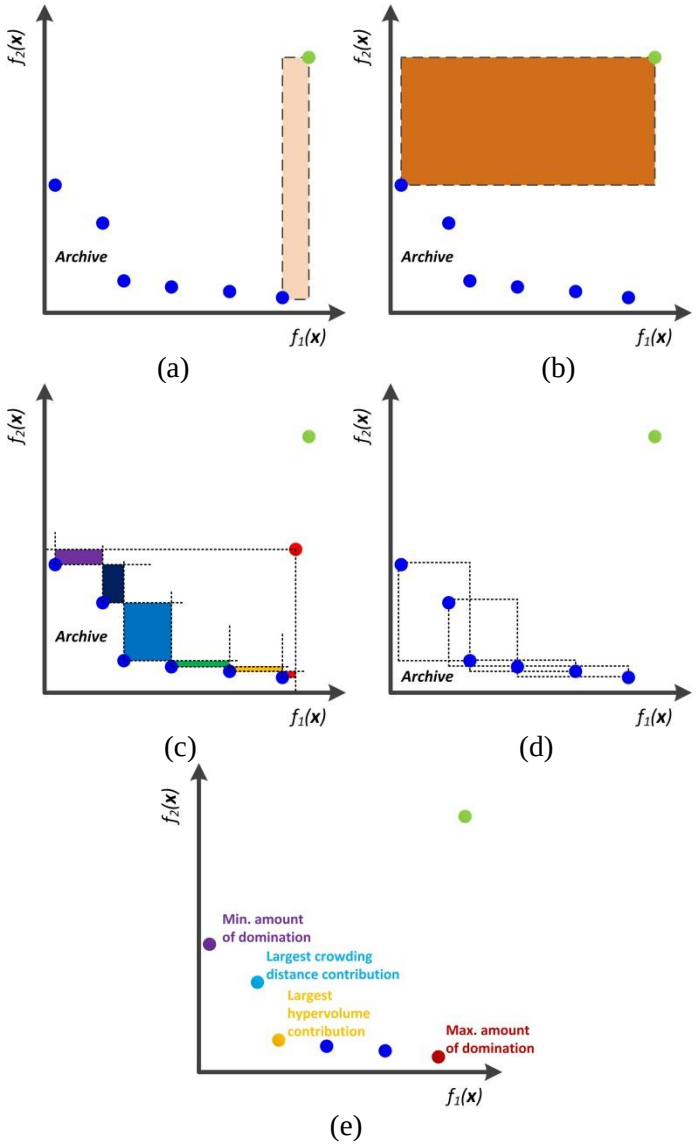


Figure 4.1 Examples of solutions selected by the four low-level heuristics

(2) The solution in the *Archive* with the maximum amount of domination. The second strategy is defined similarly to 1). For $\forall \mathbf{x} \in Archive$ that dominates the *new solution*,

$$\begin{aligned} \mathbf{x}_{select} &= \arg\max_{\mathbf{x}} (\Delta dom_{\mathbf{x},\mathbf{x}_{new}}) \\ &= \arg\max_{\mathbf{x}} \left(\prod_{i=1, f_i(\mathbf{x}) \neq f_i(\mathbf{x}_{new})}^{M} (|f_i(\mathbf{x}) - f_i(\mathbf{x}_{new})| / R_i) \right) \end{aligned} \tag{4.10}$$

The only difference is that this time the solution of the maximum domination amount compared to the *new solution* will be chosen. The strategy emphasizes more on the exploitation of better neighboring solutions compared to strategy (1) as (1) aims to maintain a balance between exploration and exploitation. The first two strategies are *new solution* dependent. Next we will introduce two new solution independent strategies. As shown in Figure 4.1(b), the selected solution using the second strategy dominates *current solution* the most.

(3) The solution in the *Archive* with the largest hypervolume (HV) contribution. In (3) we compute the hypervolume contribution of each point in *Archive* using the method proposed by Emmerich et al (2005). Hypervolume contribution quantifies how much each point in the Pareto front contributes to the HV as explained in Figure 4.1(c); the areas of the colored rectangles indicate the hypervolume contribution for each solution in the *Archive*. A large value of HV contribution indicates that the point stays in a less explored portion of the Pareto front whilst maintaining good convergent performance.

(4) The solution in the *Archive* with the largest crowding distance. The last strategy makes use of the technique called crowding distance (Deb et al, 2002). The point with the largest crowding distance will be selected. The strategy is inclined to exploration (diversity) in the EvE dilemma. As can be seen in Figure 4.1(d), in the minimization case, the crowding distance for each solution in the *Archive* is determined by the area of the bounding box formed by its adjacent solutions.

As illustrated in Figure 4.2(e) which gives a comparison of the solutions selected by the proposed four low-level heuristics, each low-level heuristic designed has its own emphasis and intention. The hyper-heuristic scheme is designed to adaptively switch between different priorities that suits current search endeavor the best, and therefore could be applied to tackle different instances without further modification.

Figure 4.2 depicts the overall mechanism of MOSA/R and the co-acting hyper-heuristic in a flowchart.

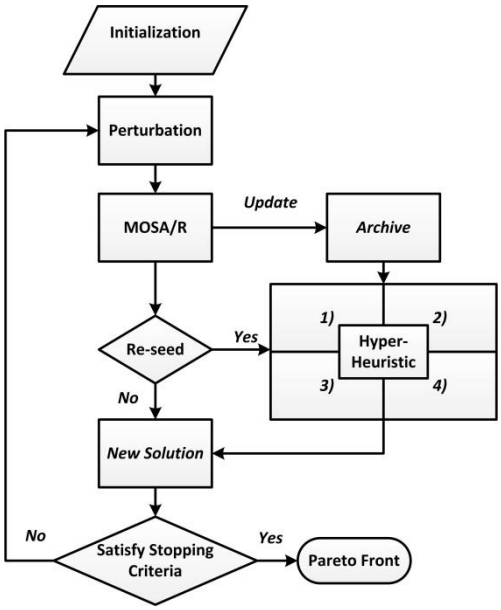


Figure 4.2 Flowchart of MOSA/R and embedded hyper-heuristic

4.4. Benchmark case studies

4.4.1. Test cases

Here we apply four algorithms, the proposed algorithm MOSA/R-HH, an advanced multi-objective simulated annealing algorithm AMOSA (Bandyopadhyay et al, 2008), a fast and elitist multi-objective genetic algorithm NSGA-II (Deb et al, 2002), and A multi-objective evolutionary algorithm based on decomposition MOEA/D (Zhang and Li, 2007), to 14 benchmark test problems from DTLZ (Deb et al, 2002) and UF (Zhang et al., 2008) test suites. The three algorithms adopted for comparison are among the most recognized multi-objective algorithms and have been applied for a variety of optimization porblems. The test sets are considered to be representative due to their diverse properties as listed in Table 4.2. All algorithms will be executed 5 times independently for each test problem.

Table 4.2 Benchmark test problem properties

Instance	*# Obj.*	*# Var.*	*Properties*
DTLZ1	3	6	Linear Pareto, multimodal
DTLZ2	3	7	Concave Pareto
DTLZ3	3	10	Concave Pareto, multimodal
DTLZ4	3	10	Concave Pareto, biased solutions distribution
DTLZ5	3	10	Concave degenerated Pareto

DTLZ6	3	10	Concave Pareto, biased solutions distribution
DTLZ7	3	10	Discontinuous Pareto
UF1	2	10	Convex Pareto
UF2	2	10	Convex Pareto
UF3	2	10	Convex Pareto
UF4	2	10	Concave Pareto
UF5	2	10	Discrete Pareto
UF6	2	10	Discontinuous Pareto,
UF7	2	10	Linear Pareto

4.4.2. Parametric setting

The initial temperature is determined that virtually all solutions are accepted at the beginning 'burn in' period (Suman and Kumar, 2006). The stopping criterion, i.e., the final temperature is chosen to control the error. In this research, the starting temperature T_{max} and final temperature T_{min} values of AMOSA and MOSA/R-HH are set to be 100 and 10^{-5}, respectively. The total number of iterations, denoted as *iter*, is chosen to be 20,000 for DTLZ1 and DTLZ2, 30,000 for DTLZ3-7, and 100,000 for UF test instances. For temperature decrement $T = \Phi(T)$, we adopt the exponential approach,

$$T_{i+1} = \alpha^i T_i \tag{4.11}$$

where $0 < \alpha < 1$ is chosen to be 0.8. Note that each parameter in AMOSA is set to be the same as that of MOSA/R-HH.

For NSGA-II and MOEA/D, the total number of function evaluations is set in accordance with AMOSA and MOSA/R-HH. Other parameters used follow those used in literature (Deb et al, 2002; Zhang and Li, 2007). For 2-objective test problems, the population size is set to be 150, and 300 for 3-objective test problems. The distribution indices of Simulated Crossover (SBX) and the polynomial mutation are set to be 20. The crossover rate is 1.00 and the mutation ration is 1/*n* where *n* is the length of decision vector. In MOEA/D, Tchebycheff approach is used and the size of neighbor population is set to be 20. All initial solutions are generated randomly form the decision space of the problems.

4.4.3. Performance metrics

For multi-objective optimization (MOO), an algorithm should provide a set of solutions that realize the optimal trade-offs between the considered optimization objectives, i.e., Pareto set. Therefore, the performance comparison of MOO algorithms is based on their Pareto sets. In this study, two popular metrics IGD and HV are used to quantify the performance of the algorithms.

Inverted Generational Distance (IGD)

The IGD indicator measures the degree of convergence by computing the average of the minimum distance of points in the true Pareto front (*PF**) to points in Pareto front obtained (*PF*), as described below,

$$IGD(PF, PF^*) = \frac{\sum_{\mathbf{f}^* \in PF^*,\, i=1}^{|PF^*|} \sqrt{\min_{\mathbf{f} \in PF} \left(\sum_{m=1}^{M} (f_m^i * - f_m)^2 \right)}}{|PF^*|} \tag{4.12}$$

where M is the number of objectives, f_m is the m-th objective value of $\mathbf{f} \in PF$. In Equation (4.12), $\min_{\mathbf{f} \in PF} \left(\sum_{m=1}^{M} (f_m^i * - f_m)^2 \right)$ calculates the minimum Euclidean distance between the ith point in *PF** and points in *PF*. A lower value of IGD indicates better convergence and completeness of the *PF* obtained.

Hypervolume (HV)

Refer to Equation (4.5). HV indicator measures convergence as well as diversity. The calculation of HV requires normalized objective function values and in this paper HV stands for the percentage covered by the Pareto front of the cuboid defined by the reference point and the original point (0, 0, 0). As mentioned earlier, the reference point is set to be 1.1 times the upper bound of the *PF**.

4.4.4. Results and discussions

The benchmark experiment examines the performance of MOSA/R-HH, AMOSA, NSGA-II, and MOEA/D as applied to DTLZ and UF test suites. The analysis results are based on 5 independent test runs. The mean and standard deviation of IGD and HV are recorded. All computations are carried out within MATLAB on a 2.40GHz Xeon E5620 desktop.

Tables 4.3 and 4.4 illustrate the relative performance of all four algorithms in terms of the two metrics IGD and HV where we keep 4 significant digits for mean and 3 significant figures for standard deviation. The shaded grids indicate the best result in each test in terms of the mean value. The performance comparison as well as the robustness of each algorithm are also illustrated in Figures 4.4. As can be observed from the figure, MOSA/R-HH prevails in DTLZ1, DTLZ2, DTLZ5 and DTLZ7 in both metrics. MOEA/D has an edge over MOSA/R-HH in DTLZ3, while MOSA/R-HH performs significantly better than NSGA-II and AMOSA. DTLZ4 is a close race for MOSA/R-HH, NSGA-II and MOEA/D. And for DTLZ6, MOSA/R-HH, AMOSA and MOEA/D all demonstrate similar performance. Figure 4.5 depicts the Pareto surface obtained by each algorithm when applied to DTLZ1 test case. For UF test cases, MOSA/R-HH takes the lead in three of them in both IGD and HV, which is the best among the four algorithms. Figure 4.6 shows an example of the Pareto front obtained by each algorithm for UF4 in comparison with the true Pareto front. It can be noticed that the Pareto front obtained by MOSA/R-HH stays close to the true Pareto front and maintains good diversity. The performance of AMOSA, NAGA-II and MOEA/D fluctuate as test instance changes due to different problem properties. On the other hand, MOSA/R-HH is more robust and outperforms other algorithms when tackling most test instances because of the adaptive hyper-heuristic scheme.

Table 4.3 Numerical test results: IGD mean and standard deviation

Instance	MOSA/R-HH	AMOSA	NSGA-II	MOEA/D
DTLZ1	0.007191 (3.69E-4)	0.02134 (0.00506)	1.656 (0.538)	0.01315 (0.00195)
DTLZ2	0.01403 (0.00127)	0.01992 (0.00107)	0.03093 (0.00147)	0.02434 (0.00173)
DTLZ3	0.06330 (0.00380)	0.7198 (0.131)	7.419 (1.87)	0.0342 (0.0125)
DTLZ4	0.02263 (0.00222)	0.07643 (0.00456)	0.02176 (0.000668)	0.02334 (0.00176)
DTLZ5	6.356E-4 (4.34E-5)	0.001956 (1.49E-4)	0.001390 (2.74E-4)	0.002541 (0.0966)
DTLZ6	3.231 E-4 (5.42E-6)	4.404E-4 (1.85E-4)	0.8738 (0.0762)	0.001792 (2.20E-4)
DTLZ7	0.01657 (9.49E-4)	0.01928 (5.45E-4)	0.8235 (0.0211)	0.06502 (0.00152)

UF1	0.01252 (0.00189)	0.03509 (0.00250)	0.01972 (0.00967)	0.01938 (0.00567)
UF2	0.002974 (6.25E-4)	0.005458 (8.87E-05)	0.006871 (0.00365)	0.01876 (0.00563)
UF3	0.2477 (0.104)	0.3797 (0.368)	0.1559 (0.0131)	0.2553 (0.0323)
UF4	0.01905 (8.76E-4)	0.03124 (1.99E-4)	0.03792 (0.00397)	0.04796 (0.00513)
UF5	0.1636 (0.00666)	0.1523 (0.0242)	0.6759 (0.279)	0.6501 (0.292)
UF6	0.1412 (0.0816)	0.09371 (4.34E-06)	0.4929 (0.0963)	0.5606 (0.151)
UF7	0.01713 (1.33 E-4)	0.03393 (0.00514)	0.008407 (0.00309)	0.005269 (5.043E-4)

Table 4.4 Numerical test results: HV mean and standard deviation

Instance	MOSA/R-HH	AMOSA	NSGA-II	MOEA/D
DTLZ1	0.8593 (0.0204)	0.8312 (0.0184)	0.04210 (0.0941)	0.8353 (0.0282)
DTLZ2	0.5945 (0.00586)	0.5850 (0.00130)	0.5663 (0.00832)	0.5789 (0.00420)
DTLZ3	0.5280 (0.0380)	0.004466 (0.00470)	0.001404 (0.00236)	0.5376 (0.0248)
DTLZ4	0.5739 (0.00869)	0.5535 (0.00738)	0.5686 (0.00765)	0.5763 (0.00877)
DTLZ5	0.2139 (0.00157)	0.2096 (0.00125)	0.2097 (0.00100)	0.2038 (0.00356)
DTLZ6	0.2059 (0.00568)	0.2029 (0.00166)	0.001440 (0.00211)	0.2012 (0.00119)
DTLZ7	0.2635 (0.00549)	0.2580 (0.0122)	0.1683 (0.00304)	0.2498 (0.00557)
UF1	0.7114 (0.00231)	0.683 (0.00198)	0.6958 (0.0126)	0.6962 (6.37E-4)
UF2	0.7207 (5.52 E-4)	0.71843 (4.03E-4)	0.7165 (0.00351)	0.7036 (0.00355)
UF3	0.4724 (0.0993)	0.4098 (0.227)	0.5196 (0.0204)	0.3787 (0.0454)
UF4	0.4224 (0.00295)	0.4044 (0.00659)	0.3919 (0.00760)	0.3885 (0.0131)
UF5	0.3613 (0.0346)	0.3651 (0.0405)	0.05647 0.0524	0.1128 (0.158)
UF6	0.3287 (0.0428)	0.3487 (0.00766)	0.1104 0.0413	0.2214 (0.0643)
UF7	0.5677 (0.00127)	0.5454 (0.00541)	0.5734 (0.00451)	0.5773 (0.00169)

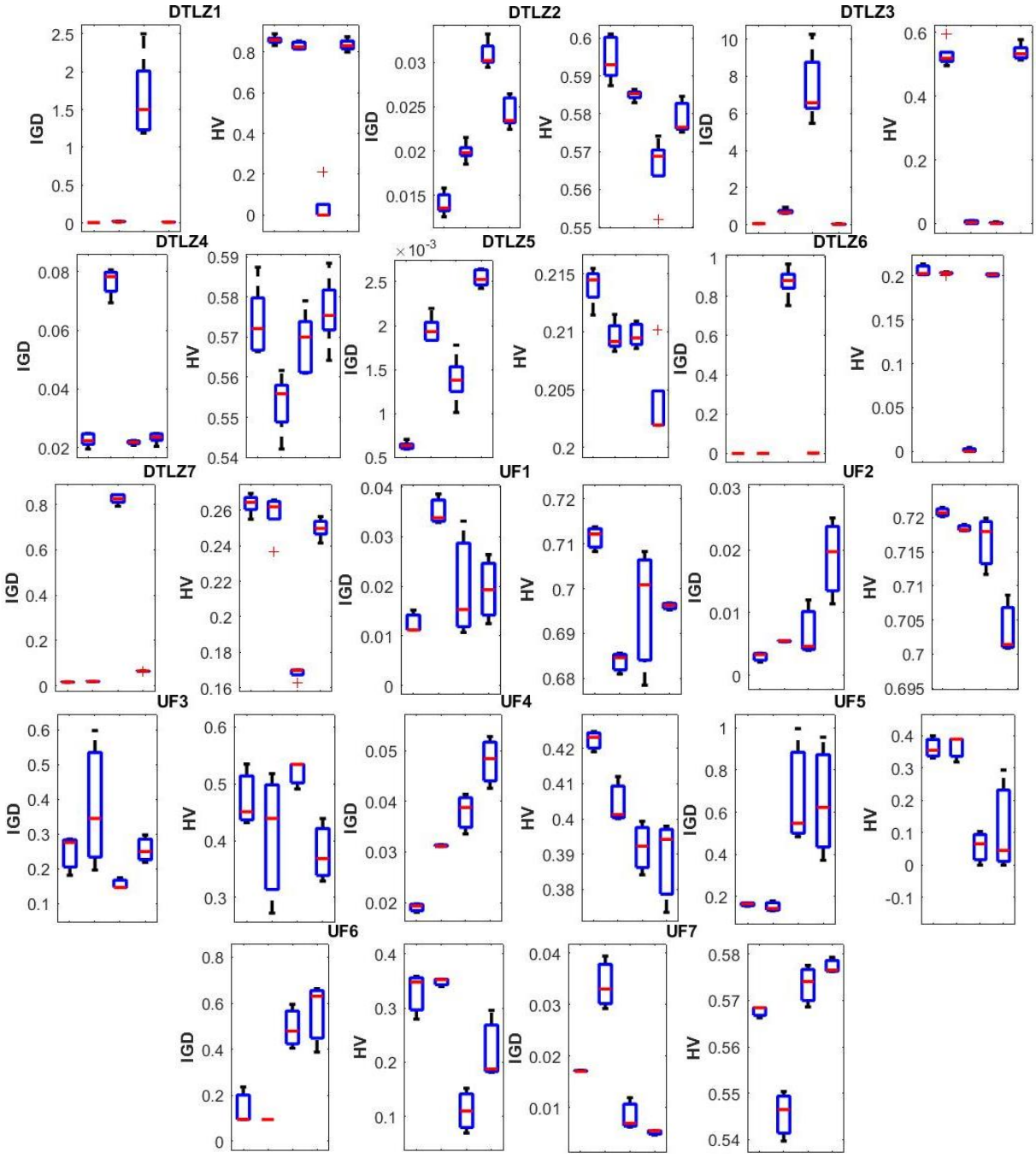
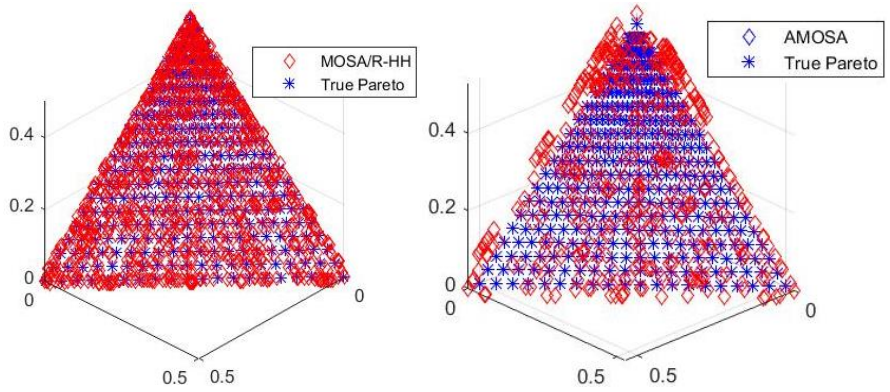
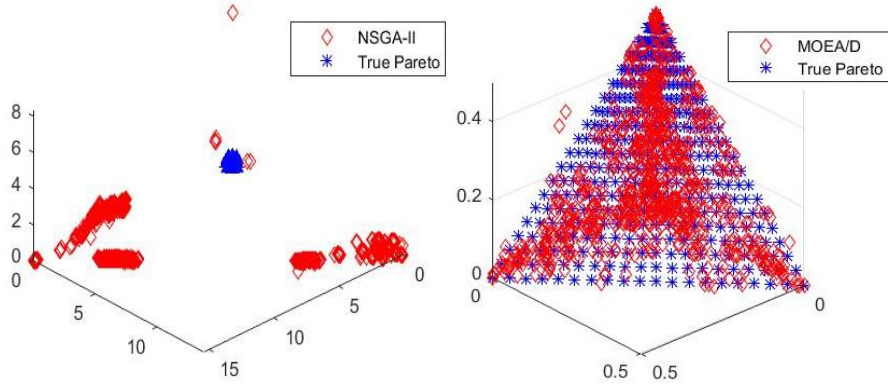


Figure 4.4 IGD and HV comparison of the four algorithms on the DTLZ and UF problems (From left to right: MOSA/R-HH, AMOSA, NAGA-II, MOEA/D)

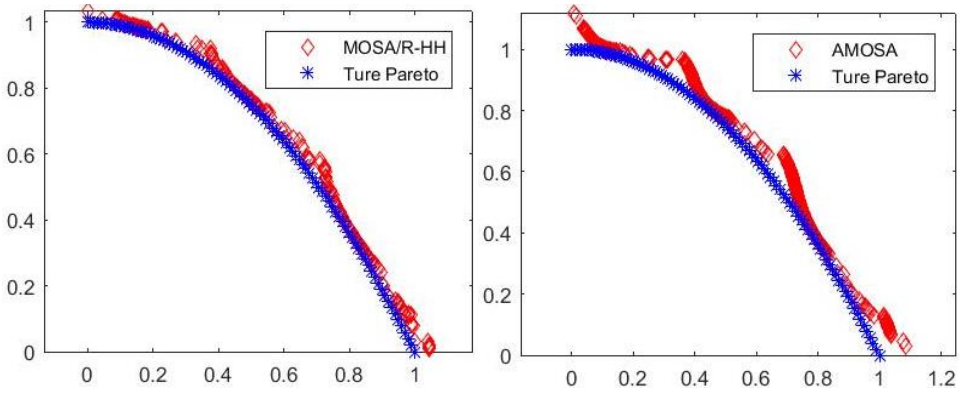


(a) MOSA/R-HH (b) AMOSA

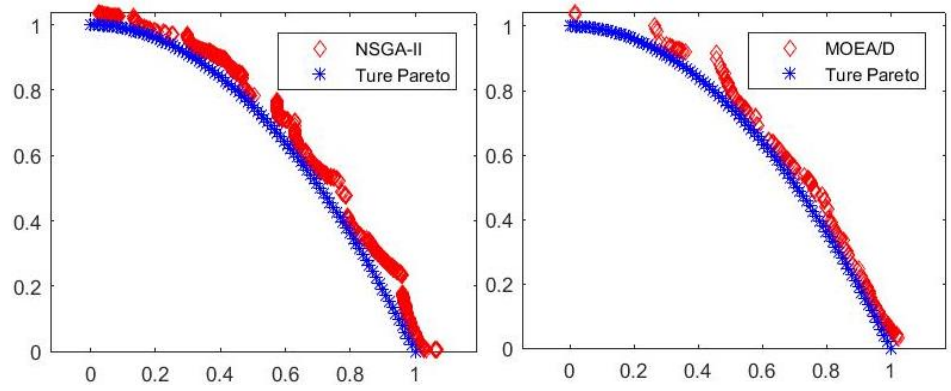


(c) NSGA-II (d) MOEA/D

Figure 4.5 Pareto front obtained by each algorithm for test instance DTLZ1



(a) MOSA/R-HH (b) AMOSA



(c) NSGA-II (d) MOEA/D

Figure 4.6 Pareto front obtained by each algorithm for test instance UF4

4.4.5. Structural fault identification using MOSA/R-HH

In this section, we apply the proposed approach (MOSA/R-HH) and the original MOSA/R to a practical engineering problem, the identification of fault parameters in a structure. Structural fault identification is generally realized by inverse analysis through comparison of sensor measurements and model prediction in the parametric space. Here we specifically utilize the vibration response measurements (Cao et al, 2018a). We aim at solving the problem using the hyper-heuristic framework developed without taking advantage of any empirical domain-knowledge.

In model-based fault identification, a credible finite element model of the structure being monitored is available. The stiffness matrix of the structure under the healthy condition is denoted as $\mathbf{K}^R = \sum_{i=1}^{n} \mathbf{K}_i^R$, where n is the number of elements, and $\mathbf{K}_i^R$ is the reference (healthy) stiffness of the i-th element. Without loss of generality, we assume that damage causes stiffness change. The stiffness matrix of the structure with fault is denoted as $\mathbf{K}^D = \sum_{i=1}^{n} \mathbf{K}_i^D$, where $\mathbf{K}_i^D = (1-\alpha_i)\mathbf{K}_i^R$. $\alpha_i \in [0, 1]$ $(i = 1, \cdots, n)$ is the fault index for the i-th element. For example, if the i-th element suffers from damage that leads to a 20% of stiffness loss, then $\alpha_i = 0.2$. We further assume that the structure is lightly damped. The j-th eigenvalue (square of natural frequency) and the j-th mode (eigenvector) are related as $\lambda_j = \{\phi_j\}^T \mathbf{K} \{\phi_j\}$. The change of the j-th eigenvalue from the healthy status to the damaged status can be derived as (Cao et al, 2018a),

$$\Delta\lambda_j = \{\phi_j\}^T (\mathbf{K}^D - \mathbf{K}^R)\{\phi_j\} = \sum_{i=1}^{n} \alpha_i \{\phi_j\}^T \mathbf{K}_i^R \{\phi_j\} \tag{4.13}$$

which can be re-written as

$$\Delta\lambda_j = \sum_{i=1}^{n} \alpha_i \cdot S_{ji} \tag{4.14}$$

or, in matrix/vector form,

$$\boldsymbol{\Delta\lambda} = \mathbf{S}\boldsymbol{\alpha} \tag{4.15}$$

where **S** is the sensitivity matrix whose elements are given in Equation (4.13), and $\boldsymbol{\Delta\lambda}$ and $\boldsymbol{\alpha}$ are, respectively, the *q*-dimensional natural frequency change vector (based on the comparison of measurements and baseline healthy results) and the *n*-dimensional fault index vector.

It is worth noting that the inverse identification problem (Equation (4.15)) is usually underdetermined in engineering practices, because *n*, the number of unknowns (i.e., the number of finite elements), is usually much greater than *q*, the number of natural frequencies that can be realistically measured. This serves as the main reason that we want to avoid matrix inversion of **S** and resort to optimization by minimizing the difference between the measurements and predictions obtained from a model with sampled fault index values. In this study, we adopt a correlation coefficient, referred to as the multiple damage location assurance criterion (MDLAC) (Messina et al, 1998; Barthorpe et al, 2017; Cao et al, 2018a), to compare two natural frequency change vectors, as expressed below,

$$\text{MDLAC}(\Delta\boldsymbol{\lambda}, \boldsymbol{\alpha}) = \frac{\left\langle \Delta\boldsymbol{\lambda}, \delta\boldsymbol{\lambda}(\boldsymbol{\alpha}) \right\rangle^2}{\left\langle \Delta\boldsymbol{\lambda}, \Delta\boldsymbol{\lambda} \right\rangle \cdot \left\langle \delta\boldsymbol{\lambda}(\boldsymbol{\alpha}), \delta\boldsymbol{\lambda}(\boldsymbol{\alpha}) \right\rangle} \tag{4.16}$$

where $\langle *, * \rangle$ calculates the inner product of two vectors. $\text{MDLAC}(\Delta\boldsymbol{\lambda}, \boldsymbol{\alpha}) \in [0,1]$ captures the similarity between measured frequency change $\boldsymbol{\Delta\lambda}$ and predicted frequency change $\delta\boldsymbol{\lambda}$. Furthermore, in addition to natural frequency change information, we also take into consideration the mode shape change information. For the *j*-th mode shape which itself is a vector, we can compare the measured change and predicted change using MDLAC in a similar manner. Therefore, a multi-objective minimization problem for an *n*-element structure can then be formulated as following,

Find: $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, ..., \alpha_n\}$

Minimize: $f_1 = -\text{MDLAC}(\Delta\boldsymbol{\lambda}, \boldsymbol{\alpha})$,

$f_2 = -\text{MDLAC}(\{\Delta\phi\}, \boldsymbol{\alpha})$

Subject to: $\alpha^l \le \alpha_i \le \alpha^u$ (4.17)

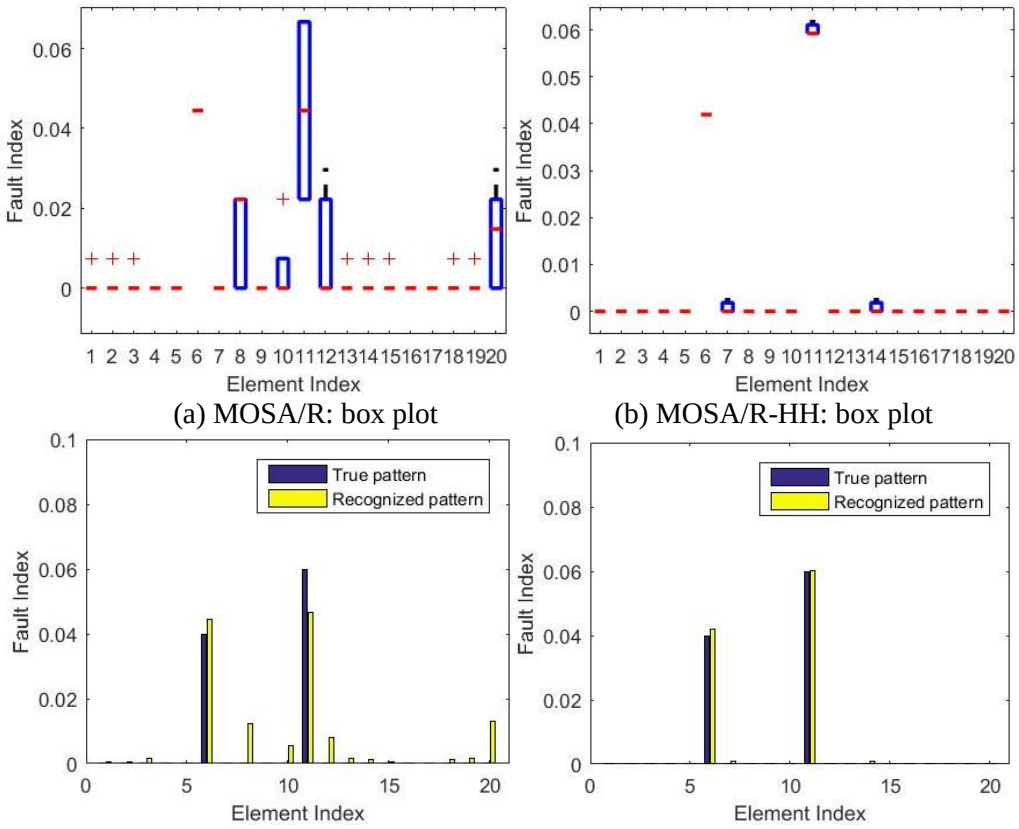
where α^l and α^u are the pre-specified lower bound and upper bound of the fault index. The optimization problem defined above is non-convex.

Prior and empirical knowledge often plays an important role when tackling this type of structural fault identification problem due to infinitely many combinations of possible fault patterns. For example, some studies assume the number of faults is known beforehand (Shuai et al, 2017). Some other investigations take advantage of the sparse nature of the fault indices (Huang et al, 2017; Cao et al, 2018b). In this study, we apply the MOSA/R-HH to identify the fault pattern in terms of $\boldsymbol{\alpha}$. We demonstrate the effectiveness of the adaptive hyper-heuristic approach, whereas we do not exploit any domain knowledge. To facilitate easy re-production of case analyses for interested readers, a benchmark cantilever beam model with varying number of elements and different fault patterns is used in this case demonstration. The Young's modulus of the beam is 69 GPa, the length per element is set as 10 m, and the area of cross-section is 1 m^3. The measurements of mode shape and natural frequencies used are simulated directly from the finite element models which are subject to 2‰ standard Gaussian uncertainties. Hereafter the measurements available to fault identification are limited to the first 5 natural frequencies and the 2nd mode shape.

4.5.1. Case study 1: 20 elements, 2 faults

We first carry out the case study on a 20-element cantilever beam. The faults are on the 6th and 11th element with severities $\alpha_6 = 0.04$ and $\alpha_{11} = 0.06$, respectively. MOSA/R-HH proposed and MOSA/R are applied without knowing the number of faults. A set of optimal candidates are obtained, owing to the tradeoff between objectives. Each solution obtained corresponds to one possible fault pattern. Figure 4.7(a) and Figure 4.7(b) show the mean and variance of the solution sets with respect to the fault index for each element, in which mean value is represented by a dash, variance is depicted as a box, and plus stands for outlier. The uncertainty and fluctuation of the results mainly come from the noise introduced to the measurements and the under-determined nature of the problem. As seen in the figures, the results of MOSA/R-HH are more robust and fewer outliers are recognized. The mean values are then compared to the true fault pattern indices in Figure 4.7(c) and Figure 4.7(d). As illustrated, MOSA-HH is able to

identify the location and severity of the fault pattern with better precision compared to MOSA/R due to the incorporated reinforcement learning hyper-heuristic. It adaptively adjusts the search direction as it progresses to yield a solution set of better distribution and accuracy.



(a) MOSA/R: box plot (b) MOSA/R-HH: box plot

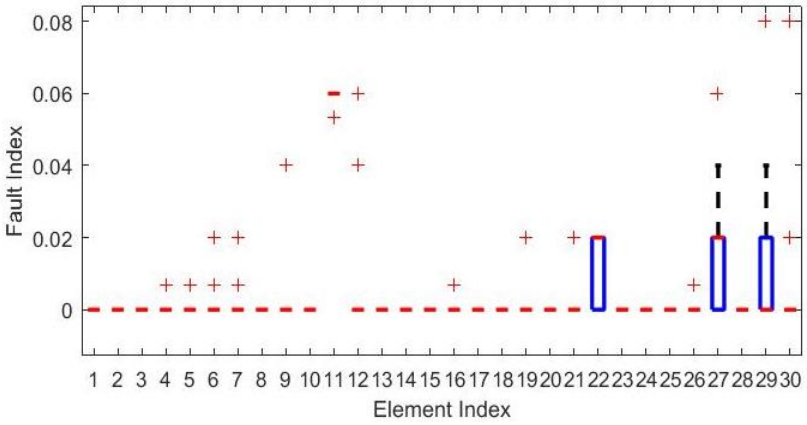
(c) MOSA/R: mean value (d) MOSA/R-HH: mean value

Figure 4.7 Case study 1: fault identification results using MOSA/R and MOSA/R-HH

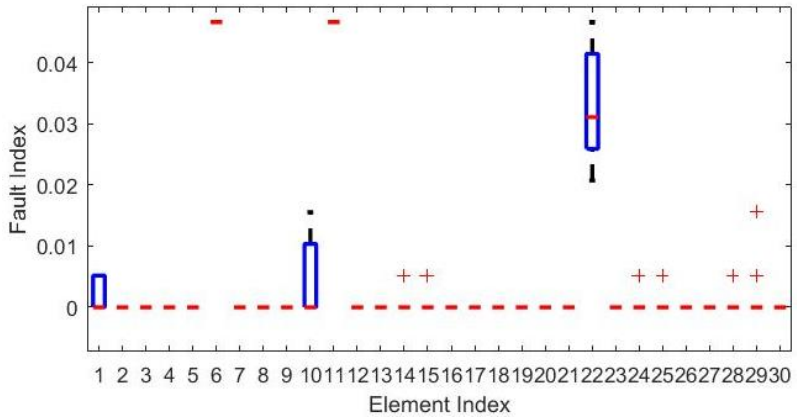
4.5.2. Case study 2: 30 elements, 3 faults

In the second case study, we perform a more difficult fault identification investigation using a 30-element cantilever beam. Three elements (6th, 11th and 22nd) are subject to faults with severities $\alpha_6 = 0.04$, $\alpha_{11} = 0.06$ and $\alpha_{22} = 0.02$ respectively. Compared to the case study conducted in Section 4.5.1, the case presented in this section is more challenging because of the many more possible combinations of fault patterns, the number of which grows exponentially with the number of elements. The proposed MOSA/R-HH is still capable of identifying a set of optimal solutions as possible fault patterns. Figure 4.8(a) and Figure 4.8(b) compare the mean and variance of the optimal solutions generated using

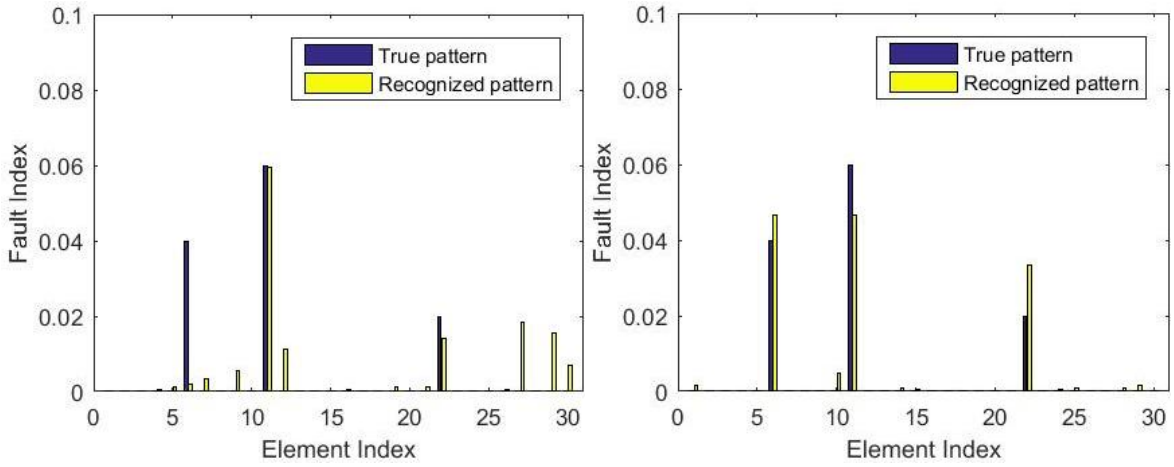
MOSA/R and MOSA/R-HH. As observed, the result set of MOSA/R-HH is more consistent and thus has fewer outliers. Due to the enlarged search space, the solutions tend to have larger variance compared to that reported in Section 4.5.1. The mean value is then compared to the true fault pattern in Figure 4.8(c) and Figure 4.8(d). MOSA/R-HH demonstrates better performance compared to MOSA/R, For an ideal model without uncertainty, adding variables (element of the structure) alone would not change the essence of the problem. In other words, if the optimization process lasts long enough, the quality of the final solutions would not deteriorate. However, errors and uncertainties are inevitable in engineering practices and also play a role in our simulation. Nevertheless, MOSA/R-HH is capable of identifying the fault pattern in terms of both location and severity while MOSA/R, in this case investigation, completely omits the fault on 6th element (Figure 4.8(c)). The mean values of the MOSA/R-HH results bear some small errors but the overall fault pattern is practically recognized without using domain knowledge.



(a) MOSA/R: box plot



(b) MOSA/R-HH: box plot



(c) MOSA/R: mean value (d) MOSA/R-HH: mean value

Figure 4.8 Case study 2: fault identification results using MOSA/R and MOSA/R-HH

4.6. Concluding remarks

In this research, we formulate a hyper-heuristic scheme that works coherently with MOSA/R towards multi-objective optimization problems, featuring domination amount, crowding distance and hypervolume calculations. The hyper-heuristic scheme can be adjusted at high-level by changing heuristic selection and credit assignment strategies or at low-level by customizing the heuristic repository to meet different optimization requirements. It can also be used to investigate the relation between heuristics and problem instances. The proposed MOSA/R-HH yields better results than other MOSA algorithm like AMOSA and representative evolutionary algorithms like NSGA-II and MOEA/D when applied to benchmark test cases. Hyper-heuristic methodology is promising as it can address the problem in different ways based on on-line learning. Finally, the proposed hyper-heuristic approach can be successfully devised to solve a representative structural fault identification problem without using any domain knowledge, as the hyper-heuristic combines the merits of different algorithms and adjust the search iteratively.

Chapter 5. Structural Damage Identification Using Piezoelectric Impedance Measurement with Sparse Inverse Analysis

The impedance/admittance measurements of a piezoelectric transducer bonded to or embedded in a host structure can be used as damage indicator. When a credible model of the healthy structure, such as the finite element model, is available, using the impedance/admittance change information as input, it is possible to identify both the location and severity of damage. The inverse analysis, however, may be under-determined as the number of unknowns in high-frequency analysis is usually large while available input information is limited. The fundamental challenge thus is how to find a small set of solutions that cover the true damage scenario. In this research we cast the damage identification problem into a multi-objective optimization framework to tackle this challenge. With damage locations and severities as unknown variables, one of the objective functions is the difference between impedance-based model prediction in the parametric space and the actual measurements. Considering that damage occurrence generally affects only a small number of elements, we choose the sparsity of the unknown variables as another objective function, deliberately, the l_0 norm. Subsequently, a multi-objective Dividing RECTangles (DIRECT) algorithm is developed to facilitate the inverse analysis where the sparsity is further emphasized by sigmoid transformation. As a deterministic technique, this approach yields results that are repeatable and conclusive. In addition, only one algorithmic parameter, the number of function evaluations, is needed. Numerical and experimental case studies demonstrate that the proposed framework is capable of obtaining high-quality damage identification solutions with limited measurement information.

5.1 Introduction

Extensive research has been conducted on structural health monitoring (SHM) to protect structures from catastrophic failures. Many structural health monitoring (SHM) systems use structural dynamic responses measured by sensors to elucidate the health condition. A traditional class of methods is vibration-based SHM, which analyzes and interprets measureable modal properties such as natural

frequencies and mode shapes to determine damage locations and severities (Kim and Stubbs, 2003; Maity and Tripathy, 2005; Jiang et al, 2006; Jassim et al, 2013; Cao et al, 2017). However, in practical situations, normally only lower-order modes with large wavelengths can be realistically excited and measured; thus, these methods may not be sensitive to small-sized damage (Kim and Wang, 2014). Another well-known class of methods is based on wave propagation, which uses the change of transient wave upon its passage through damage site to infer damage occurrence (Michaels and Michaels, 2007; Harley and Moura, 2014). While these high-frequency methods may entertain high detection sensitivity, it is generally difficult to use the transient responses to identify damage accurately, especially to quantify the severity of damage (Cawley and Simonetii, 2005). Piezoelectric transducers are frequently used in wave propagation-based SHM. The two-way electro-mechanical coupling of piezoelectric transducers has also allowed them to be used in piezoelectric impedance- or admittance-based methods (Park et al, 2003; Yang et al, 2008; Lim and Soh, 2012; Min et al, 2012; Annamdas and Radhika, 2013; Lim and Soh, 2014; Shuai et al, 2017). In these methods, frequency-swept harmonic voltage excitations are applied, and stationary wave responses are induced and sensed. As such, the impedance of the structure is coupled with that of the piezoelectric transducer. The change of piezoelectric impedance signature with respect to that under the undamaged baseline can be used as damage indicator.

When a credible, first-principle-based baseline model, such as the finite element model, of the healthy structure is available, we may be able to identify both the fault location and severity by using vibration responses or impedance/admittance responses which are stationary (Xia and Hao, 2003; Jiang et al, 2006; Wang and Tang, 2008; Zhou and Zuo, 2012; Shuai et al, 2017). In such an inverse analysis, we typically divide the structure into a number of segments and assume that each segment in the model is susceptible of damage occurrence, i.e., certain property of each segment is an unknown. The inverse analysis uses the changes of stationary responses, such as natural frequencies, mode shapes, response amplitudes, impedances or admittances, as inputs. Indeed, a linearized sensitivity matrix can be derived that links the segment property change vector with respect to the response measurement change vector. This seemingly straightforward formulation appears to be appealing. Nonetheless, how to solve for the

unknowns, i.e., damage locations and severities, remains a fundamental challenge. On one hand, the number of segments or the number of finite elements is generally very large, especially for SHM schemes utilizing high-frequency excitations such as those using piezoelectric impedance or admittance measurements where the mesh size must be smaller than the wavelength of the high-frequency responses. On the other hand, available measurement information is usually limited. For example, in impedance-based methods, only the impedance measurements around resonant peaks are truly sensitive to damage occurrence with high signal-to-noise ratio, and adding more measurement frequency points do not necessarily increase the row-rank of the sensitivity matrix. Therefore, the aforementioned inverse formulation may easily become under-determined (Kim and Wang 2014). Moreover, the inevitable measurement noise and modeling uncertainty further compound the difficulty (Shuai et al, 2017).

To avoid the direct inversion of the sensitivity matrix, alternatively, the problem of identifying damage location/severity in a finite element-based analysis can be cast into a global optimization formulation. Indeed, under the umbrella of optimization formulation, several global optimization techniques, such as particle swarm optimization (Begambre and Laier, 2009), differential evolution algorithm (Seyedpoor et al., 2015), genetic algorithm (Perera et al., 2010), and DIRECT algorithm (Cao et al, 2017), have been attempted, where possible property changes in all segments are treated as unknowns to be solved. One important feature of these optimization-based formulations is that usually only forward-analysis of the model is involved to facilitate comparison and minimization of the difference between model prediction in the parametric space and actual measurements. Nevertheless, given that the sensitivity matrix is under-determined, the fundamental mathematical challenge remains, i.e., there are in theory infinitely many solutions. Inevitably, many solutions that are different from the true damage scenario may be obtained. Certainly, one would hope to remove some or even most of these 'untrue' solutions by imposing constraints in the optimization framework. In a wider context, such ill-posed problems have been tackled in mathematical and statistical literature by invoking the sparsity condition (Tarantola, 2005; Kaipio and Somersalo, 2006). For example, compressed sensing in signal processing is facilitated by applying the sparsity constraint into a usually ill-posed optimization problem (Candes et al,

2006; Donoho 2006; Mascarenas et al, 2013). Interestingly, the sparsity condition is also applicable to damage identification problem. Specifically, in practical situation, damage occurrence is much more probable to affect only a small number of elements/segments in a finite element model of the structure. In other words, the unknown vector in the damage identification problem is sparse. However, few investigations have so far taken advantage of the inherent sparse nature of damage indices to address damage identification problems. Wang and Hao (2014) formulated a pattern recognition problem for damage identification and matched the pre-defined patterns following a compressed sensing-based scheme which uses sparsity properties and l_1 regularization on the unknown damage pattern vector. This approach cannot be easily extended to cases with many damage patterns. More recently, Huang et al (2017) adopted the sparse Bayesian learning for structural damage detection. Rather than providing a single estimate, sparse Bayesian learning provides a full posterior density function, which gives a sense of confidence of the approximation.

While it is intuitive to incorporate the sparsity condition into the aforementioned optimization problem, the formulation of an optimization problem (i.e., design objective with constrains) and the associated solution method are intertwined, especially when many unknown variables are involved. For the problem of damage identification, three questions arise, including 1) how to actually define sparsity condition mathematically, 2) how to integrate the sparsity condition into optimization, and 3) how to solve the optimization problem. In this paper, we develop a multi-objective optimization formulation that can be effectively solved by a deterministic global approach originated from the Dividing RECTangles (DIRECT) algorithm (Jones et al, 1993). The rationale is given as follows. The difference between model prediction in the parametric space and the measurement needs to be minimized for the sake of damage identification, which is one obvious objective function in the optimization problem. Although the sparsity condition can be included as a constraint, in this research we opt to treat it as a separate objective function to be minimized. First this can avoid the employment of weighting constant, which is generally ad hoc, in solving the usual constrained optimization problem. Second, this formulation naturally produces multiple solutions, which reflects the under-determined nature of the problem.

Moreover, as will be shown comprehensively later in this paper, a proper choice of sparse regularization under the multi-objective formulation (i.e., treating the sparsity as the second objective function) has the potential of yielding a *small* set of solutions that fits better the true damage scenario. In comparison, if a single composite objective function is used, it yields one single solution which however may not capture the true damage at all. It is worth emphasizing that the DIRECT algorithm is particularly suitable in solving the multi-objective optimization problem formulated. Mathematically, it is a deterministic technique, and thus the results obtained are repeatable and conclusive without ambiguities. Only one algorithmic parameter, the number of function evaluations, is needed here. To enhance the computational efficiency, a new sampling/division scheme in the unknown parametric space is established. In this research, to demonstrate its effectiveness, this new framework of damage identification is applied to piezoelectric impedance-based SHM featuring high detection sensitivity but with large number of unknown variables.

The rest of the chapter is organized as follows. In Section 5.2, we briefly outline the basic equations and background information of piezoelectric impedance-based technique, followed by the optimization problem formulation. The details of the sparse multi-objective DIRECT algorithm are presented in Section 5.3. In Section 5.4, the proposed method is evaluated through two benchmark damage scenarios. Further validations using experimental results are conducted in Section 5. Concluding remarks are given in Section 5.6.

5.2 Problem formulation

5.2.1 Piezoelectric impedance/admittance active sensing

In piezoelectric impedance/admittance-based damage detection, a piezoelectric transducer is attached to the host structure. Frequency-sweeping, harmonic excitation voltage is applied to the piezoelectric transducer, which induces structural oscillation. Owing to the two-way electro-mechanical coupling, the local structural oscillation in turn affects the electrical response of the transducer. As such, the impedance/admittance of the transducer is directly related to the impedance of the underlying structure.

Indeed, the equations of motion of the coupled sensor-structure interaction system in the finite element form can be derived as (Wang and Tang, 2010),

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} + \mathbf{K}_{12}Q = \mathbf{0} \tag{5.1a}$$

$$K_c Q + \mathbf{K}_{12}^{\mathbf{T}}\mathbf{q} = V_{\text{in}} \tag{5.1b}$$

where $\mathbf{q}$ is the structural displacement vector, $\mathbf{M}$, $\mathbf{K}$ and $\mathbf{C}$ are the mass matrix, stiffness matrix and damping matrix, respectively, $\mathbf{K}_{12}$ is the vector indicating the electro-mechanical coupling due to piezoelectric effect, K_c is the reciprocal of the capacitance of the piezoelectric transducer, Q is the electrical charge on the surface of the piezoelectric transducer, and V_{in} is the excitation voltage. Under harmonic excitation, Equations (5.1a) and (5.1b) can be transferred to frequency domain. The admittance of the piezoelectric transducer that is coupled to the host mechanical structure can be expressed as

$$Y(\omega) = \frac{\dot{Q}}{V_{\text{in}}} = \frac{\omega \mathbf{i}}{K_c - \mathbf{K}_{12}^T(\mathbf{K} - \mathbf{M}\omega^2 + \mathbf{C}\omega \mathbf{i})^{-1}\mathbf{K}_{12}} \tag{5.2}$$

where ω is the excitation frequency and i is the imaginary unit. While the admittance and the impedance are reciprocal with each other, here in this research without loss of generality we employ the piezoelectric admittance as the information carrier.

In discretized model-based structural health monitoring, structural damage is frequently assumed as local property change, e.g., local stiffness loss. We divide the host structure into n segments and let $\mathbf{k}_{hj}$ be the stiffness matrix of the healthy, j-th segment. The stiffness matrix of the structure with damage is then expressed as

$$\mathbf{K}_d = \sum_{j=1}^{n} \mathbf{k}_{hj}(1 - \alpha_j) \tag{5.3}$$

where $\alpha_j \in [0, 1]$ is the damage index indicating the stiffness loss in the j-th segment. For example, if the j-th segment suffers from damage that leads to a 20% stiffness loss, $\alpha_j = 0.2$. In practical situation, multiple segments may be subjected to damage or stiffness losses. Let $\boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_n]^T$ be the damage

index vector. Multiple vector elements in $\boldsymbol{\alpha}$ may be non-zero. Meanwhile, damage effect is reflected in the change of admittance of the piezoelectric transducer,

$$Y_d(\omega) = \frac{\dot{Q}_d}{V_{\text{in}}} = \frac{\omega i}{K_c - \mathbf{K}_{12}^T(\mathbf{K}_d - \mathbf{M}\omega^2 + \mathbf{C}\omega i)^{-1}\mathbf{K}_{12}} \tag{5.4}$$

Typically, the measured admittance of the damaged structure is compared with the baseline admittance information in order to identify damage location and severity.

It can be observed from Equations (5.3) and (5.4) that the admittance change is not linearly dependent upon the damage index vector. When the size of damage is small, which is the usual situation, we can use Taylor series expansion to develop a linearized relationship between the admittance change and the damage index vector (Shuai et al, 2017). The admittance of the damaged structure can be expressed as,

$$Y_d(\boldsymbol{\alpha}) \approx Y(\boldsymbol{\alpha} = \mathbf{0}) + \sum_{j=1}^{n} \frac{\partial Y}{\partial \alpha_j}|_{\alpha_j=0}\, \alpha_j \tag{5.5}$$

where

$$\frac{\partial Y}{\partial \alpha_j}|_{\alpha_j=0} = \omega i[k_c - \mathbf{K}_{12}^T(\mathbf{K} - \mathbf{M}\omega^2 + \mathbf{C}\omega i)^{-1}\mathbf{K}_{12}]^{-2}\mathbf{K}_{12}^T[\frac{\partial(\mathbf{K}_d - \mathbf{M}\omega^2 + \mathbf{C}\omega i)^{-1}}{\partial \alpha_j}|_{\alpha_j=0}]\mathbf{K}_{12} \tag{5.6}$$

The change of admittance due to damage can then be written as,

$$\begin{aligned}\Delta Y(\omega) &= Y_d - Y(\boldsymbol{\alpha} = \mathbf{0}) \\ &= \sum_{j=1}^{n}[\omega i(k_c - \mathbf{K}_{12}^T(\mathbf{K} - \mathbf{M}\omega^2 + i\omega\mathbf{C})^{-1}\mathbf{K}_{12})^{-2}\mathbf{K}_{12}^T(\mathbf{K} - \mathbf{M}\omega^2 + i\omega\mathbf{C})^{-1}\mathbf{k}_{hj}(\mathbf{K} - \mathbf{M}\omega^2 + i\omega\mathbf{C})^{-1}\mathbf{K}_{12}]\alpha_j\end{aligned} \tag{5.7}$$

Recall $\mathbf{k}_{hj}$ is the stiffness matrix of j-th (j = 1, …, n) segment when the structure is intact. Equation (5.7) gives the relationship between the admittance change and the damage index at one specific excitation frequency point ω. In impedance/admittance-based damage identification, while harmonic voltage excitation is used for active sensing, the frequency of the excitation is swept through a certain range that covers a number of structural resonances around which measurements are taken. Assume admittance values are measured at m frequency points, ω_1, ⋯, ω_m. For each frequency point, Equation (5.7) holds whereas the damage index vector remains to be the same. We then have

$$\Delta\mathbf{Y} = \begin{bmatrix} \Delta Y(\omega_1) \\ \vdots \\ \Delta Y(\omega_m) \end{bmatrix} = \mathbf{S}_{m\times n}\boldsymbol{\alpha} \tag{5.8}$$

where $\Delta\mathbf{Y}$, an *m*-dimensional vector, is the admittance changes measured at *m* frequency points, $\boldsymbol{\alpha}$ is the *n*-dimensional vector of the unknown damage indices, and $\mathbf{S}_{m\times n}$ is the sensitivity matrix.

5.2.2 Inverse analysis formulated as an optimization problem with l_0 sparse regularization

The high detection sensitivity of impedance/admittance-based active sensing approach is built upon the high-frequency responses excited and measured. A very large number of finite elements are needed to establish the baseline model for credible prediction of high-frequency responses. As we divide the structure into segments to facilitate damage identification, the structural properties of each segment remains to be identified because each segment is susceptible of fault occurrence, which yields a large number of unknowns. On the other hand, structural faults generally manifest themselves around the peaks of the piezoelectric impedance/admittance curves only, which means the input measurement information is usually limited in practice. Moreover, it is mathematically difficult to select frequency points to ensure the full rank of the sensitivity matrix (that relates fault location/severity with measurement, shown in Equation (5.8)) even if the number of frequency points is large. Therefore, the inverse identification formulation typically is under-determined (Kim and Wang, 2014). Although one may apply artificial constraints to seek for such as least square solutions, these solutions may not reflect the true fault scenario.

Hereafter, we cast the inverse identification problem into an optimization framework. Let $\Delta\mathbf{Y}$ be the measured admittance change. The prediction of admittance change in the parametric space is denoted as $\Delta\hat{\mathbf{Y}} = \mathbf{S}\hat{\boldsymbol{\alpha}}$. Certainly, we need to minimize the difference between these two, i.e.,

$$\min \left\| \mathbf{S}\hat{\boldsymbol{\alpha}} - \Delta\mathbf{Y} \right\|_2 \tag{5.9}$$

where $\left\| \bullet \right\|_p$ denotes the l_p norm defined as $\left\| \mathbf{x} \right\|_p = \left(\sum_i |x_i|^p \right)^{1/p}$. Equation (5.9) has a large number of local minima given that $\mathbf{S}$ is rank-deficient. It is worth noting that a true damage scenario in practical

situation usually affect only a small number of segments. In other words, the damage index vector $\boldsymbol{\alpha}$ is sparse by nature. Here we introduce the sparse regularization by enforcing a sparse constraint on $\hat{\boldsymbol{\alpha}}$. Traditionally, an l_2 regularizer $\|\hat{\boldsymbol{\alpha}}\|_2$ has been used (Donoho, 2006), which, however, is not designed toward sparse solutions. As illustrated in Figure 1, larger p (subscript of l_p) tends to spread out the error more evenly among variables and return a non-sparse $\hat{\boldsymbol{\alpha}}$ with many nonzero elements. Thus, we are more interested in l_0 or l_1 norm. While l_1 norm is usually used in statistical and signal processing as a convex approximation of non-convex l_0 to improve computational efficiency (Davenport et al, 2011), it introduces a mismatch between the goal of itself and that of l_0 norm (Wipf and Rao, 2004). As such, we may fail to recover the maximally sparse solution regardless of the initialization. In damage identification, finding a solution of Equation (5.9) with the presence of modeling and measurement error cannot be simply solved as a convex optimization problem because it is indeed a multi-modal problem (having multiple local optima). Therefore, the convex approximation of l_0 would not be a necessity. Thus, the second objective function here is chosen as the minimization of the l_0 norm of $\hat{\boldsymbol{\alpha}}$, i.e.,

$$\min \|\hat{\boldsymbol{\alpha}}\|_0 \tag{5.10}$$

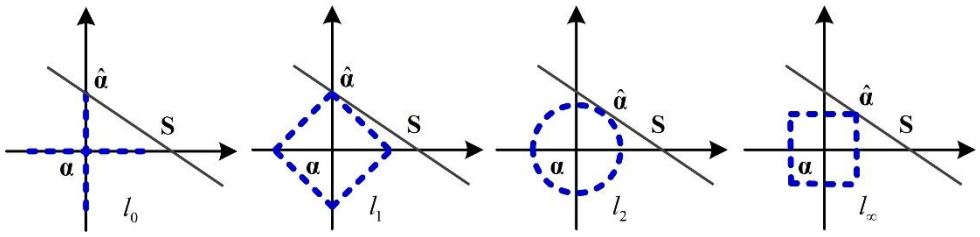


Figure 5.1. Approximation of point $\boldsymbol{\alpha}$ in $\mathbf{R}^2$ using l_0, l_1, l_2 and l_∞ norm.

One way of handling the two objective functions selected above (Equations (5.9) and (5.10)) is to formulate a composite objective function, i.e. (Cao et al, 2016),

$$\min \ \|\mathbf{S}\hat{\boldsymbol{\alpha}} - \Delta\mathbf{Y}\|_2 + \lambda\|\hat{\boldsymbol{\alpha}}\|_0 \tag{5.11}$$

where λ is the weighting factor. The choice of weights is usually ad-hoc since the relative importance of each objective is unknown. A more critical issue is that such a single objective optimization usually gives one single optimum which may or may not fit the true damage scenario at all. Indeed, mathematically, Equation (5.8) is oftentimes under-determined. While we know the damage index vector must be sparse, we generally do not know how sparse it is. In this research, we formulate a multi-objective optimization,

$$\text{Find: } \hat{\boldsymbol{\alpha}} = \{\hat{\alpha}_1, \hat{\alpha}_2, ..., \hat{\alpha}_n\}, \qquad \alpha^l \le \hat{\alpha}_j \le \alpha^u, \qquad j = 1, 2, ..., n$$

$$\text{Minimize: } f_1 = \left\|\mathbf{S}\hat{\boldsymbol{\alpha}} - \Delta\mathbf{Y}\right\|_2 \text{ and } f_2 = \left\|\hat{\boldsymbol{\alpha}}\right\|_0 \tag{5.12a, b}$$

where α^l and α^u are the lower bound and upper bound of the damage index. A fundamental advantage of this multi-objective optimization formulation is that it naturally yields a set of optimal solutions explicitly exhibiting the tradeoff between objectives, i.e., the Pareto front/surface (Deb et al, 2002). This fits exactly the under-determined nature of the damage identification problem, and provides identification results that can be used for further inspection and prognosis which is the actual procedure of performing SHM.

5.3. Damage identification using multi-objective DIRECT

5.3.1 DIRECT algorithm

DIRECT algorithm is a deterministic global optimization technique originally formulated by Jones et al (1993). The algorithm facilitates optimization tasks by Dividing Rectangles, from which its name is derived. Inherently, as a deterministic approach, DIRECT converges faster than stochastic global techniques. Moreover, while the performance of most stochastic global optimization methods depend heavily on algorithmic parameters, DIRECT has much less tunable parameters. These features are desirable for tackling damage identification problems. The original DIRECT algorithm is summarized in four steps as follows.

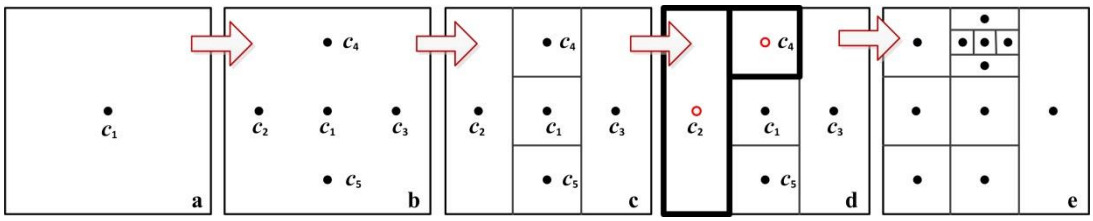


Figure 5.2. Sampling and dividing of the decision space.

1) Normalize. Normalize the decision domain to a unit cube/hyper-cube (Figure 2(a)).

$$\bar{\Omega} = \{x \in R^N : 0 \le x_i \le 1\}, \qquad i = 1, 2, ..., n \tag{5.13}$$

2) Sample and divide. Sample the center point c_1 of the cube/hyper-cube. Then evaluate the objective function values.

$$\mathbf{c}_1 \pm \mu\vec{e}_i, \qquad i = 1, 2, ..., n \tag{5.14}$$

where μ is one-third of the length of the cube, and $\vec{e}_i$ is the *i*-th unit vector in Euclidean space (Figure 5.2(b)). Dimension *i* will be divided first if *i* satisfies,

$$\arg\min_{i=1,2,...,n}\left(\min(f(\mathbf{c}_1 + \mu\vec{e}_i),\ f(\mathbf{c}_1 - \mu\vec{e}_i))\right) \tag{5.15}$$

For example, if $\min(f(\mathbf{c}_2), f(\mathbf{c}_3)) < \min(f(\mathbf{c}_4), f(\mathbf{c}_5))$ (Figure 5.2(c)), the dimension along $\mathbf{c}_2, \mathbf{c}_3$ direction will be divided first.

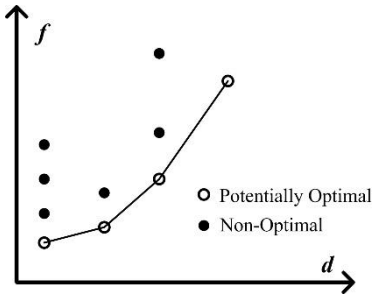


Figure 5.3. Determine the potentially optimal rectangles.

3) Determine the potentially optimal rectangles. A rectangle *j* is said to be potentially optimal if some rate-of-change constant $K > 0$ and a positive constant ε exists satisfies,

$$f(\mathbf{c}_j) - Kd_j \le f(\mathbf{c}_i) - Kd_i, \quad \text{for } \forall i \tag{5.16a}$$

$$f(\mathbf{c}_j) - Kd_j \le f_{\min} - \varepsilon \left| f_{\min} \right| \tag{5.16b}$$

where d_j measures the size of the *j*-th rectangle. In the original formulation of DIRECT, *d* is the distance from rectangle center *c* to the vertex. In Figure 5.2(d), the rectangles containing $\mathbf{c}_2$ and $\mathbf{c}_4$ are potentially optimal and thus are further divided as shown in Figure 5.2(e).

4) Iterate. Repeat Step 2 and Step 3 until maximum number of evaluation is reached. In Figure 5.2, a 2D example is provided. Higher dimensional cases could be divided and sampled following the similar procedure (Jones et al, 1993; Gablonsky, 2001).

5.3.2 Multi-objective DIRECT

The original DIRECT is designed for single-objective optimization. In this sub-section, aiming at solving the multi-objective optimization problem defined in Equation (5.12), we devise a multi-objective DIRECT approach. There have been some recent multi-objective variants of DIRECT (Wang et al, 2008; Al-Dujaili and Suresh, 2016; Wong et al, 2016). In the approach suggested by Wang et al (2008), rank strategy was used to rank the rectangles into fronts in terms of their objective values, and then $f(c)$ in Equation (5.16) was replaced with the rank to identify the potentially optimal rectangles. However, this approach only explores the rectangles on lower-right convex front (Figure 5.4) and overlooks a portion of rectangles that may lead to optimal solutions. Al-Dujaili and Suresh (2016) incorporated *d* as an additional objective value and obtained the potentially optimal rectangles by calculating the non-dominated Pareto front. One drawback of such a technique is that it expands the definition of potentially optimal to the extent where most rectangles are considered potentially optimal while only a small portion of them are rectangles of interest. Wong et al (2016) replaced *R* in Figure 4 with the hypervolume indicator and selected rectangles on the upper right Pareto front in the *hypervolume-d* plane. This approach is computationally prohibitive because hypervolume needs to be calculated iteratively, whereas hypervolume calculation is a computationally expensive NP-hard task itself.

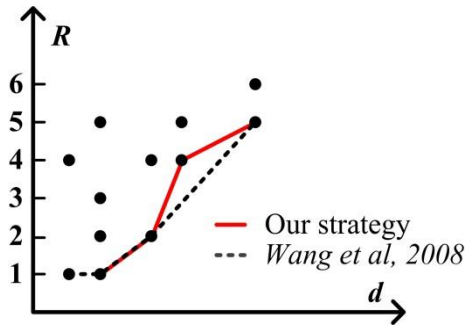


Figure 5.4. Determine the potentially optimal rectangles based on rank index R (*d:* rectangle size).

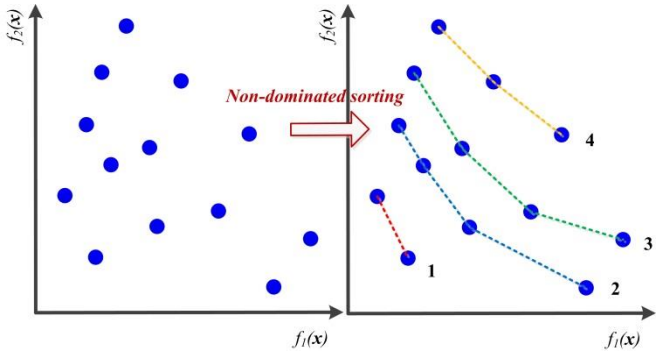


Figure 5.5. Rank based on objective values.

In this research, we employ a new strategy to fulfill the multi-objective capability and meanwhile address the aforementioned shortcomings. To determine the potentially optimal rectangles, the rectangles are ranked into fronts/surfaces using non-dominated sorting (Deb et al, 2002) as illustrated in Figure 5.5. By definition, the first front has a rank index of 1, the second front has a rank index of 2, and so on. Then the potentially optimal rectangles can be attained by projecting the rectangles to *R*-*d* plane and extract the rectangles on the lower right Pareto front as shown in Figure 5.4. In the case of *h*-objective minimization, a rectangle *j* is said to be potentially optimal if

$$R\Big(R\big(f_1(c_j),\ \ f_2(c_j),\ldots,f_h(c_j)\big),-d\Big)=1 \tag{5.17}$$

where $R(\bullet)$ is an non-dominated sorting operator that returns the rank index. In the original implementation of DIRECT, *d* is the distance measured between the rectangle center and vertex. In this research, *d* is set as the length of the longest side of the rectangle such that the algorithm groups more rectangles at the same size (Finkel, 2005). Compared to the original DIRECT algorithm, the second condition (Equation 5.16(b)) is omitted because by considering the rectangles on lower right Pareto front,

the rectangles on lower right convex hull that have the same *R* value but smaller *d* value will be naturally eliminated, which entertains the same effect of Equation 5.16(b). In a prior study, the proposed strategy has been compared to the above-mentioned three contemporary multi-objective DIRECT techniques, and exhibits favorable performance (Cao et al, 2017).

5.3.3 DIRECT for sparse exploitation

In this sub-section, we discuss the advantages of using DIRECT algorithm and the sigmoid transformation for finding sparse solutions for damage identification problem. For an *n*-dimensional optimization problem, the center $\mathbf{c}_1$ of the normalized decision space $\bar{\Omega}$ can be represented by a vector $\left[c_{1,1}, c_{1,2}, c_{1,3} \ldots c_{1,n}\right]^{\mathrm{T}}$. The neighboring decision space of $\mathbf{c}_1$ will be explored first since DIRECT divides along each dimension concurrently to identify the possibility of improvement (Figure 5.6). Therefore, if the optimal solution stays within the vicinity of the initial center, the solution can be approximated in certain number of steps due to the deterministic nature of DIRECT. In other words, if the sparsity of the solution is known *a priori*, the computational efficiency can be improved tremendously by adjusting the decision space so the center of the normalized decision space $\bar{\Omega}$ is a 0-sparse vector, i.e., zero vector. This is especially crucial when dealing with high dimensional damage identification problems where the computational cost for a global optimization is prohibitive. Hence, incorporating the fact that the damage index vector $\boldsymbol{\alpha}$ must be sparse into the problem formulation not only is necessary but also can facilitate the computationally efficient implementation of multi-objective DIRECT.

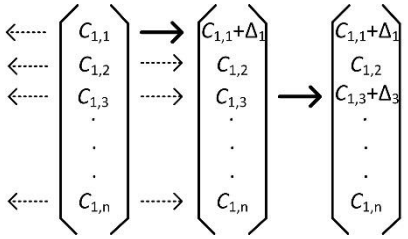


Figure 5.6. Neighboring exploration of DIRECT.

Recall Equation (5.13). The decision space of an *n*-dimensional damage identification problem can be denoted as $\alpha^l \le \hat{\alpha}_j \le \alpha^u$, $j = 1, 2, ..., n$. In order to make $\left[c_{1,1}, c_{1,2}, c_{1,3} \ldots c_{1,n}\right]^{\mathrm{T}}$ a zero vector, we need

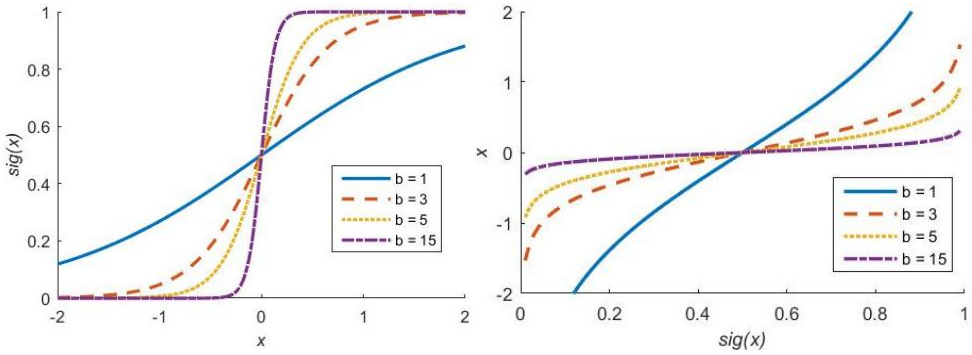
to have $-\alpha^l = \alpha^u$. As explained in Section 5.2, $\alpha_j \in [-1,1]$ is the damage index indicating the stiffness loss (+) or gain (-) in one segment. Naturally, α^l and α^u can be chosen as -1 and 1, respectively. However, actual damage is usually very small stiffness losses, and thus the choice of -1 and 1 may lead to unnecessary computational burden. To direct the algorithm to search within a small neighboring space of a sparse vector, we adopt the following sigmoid function,

$$\text{sig}(x) = \frac{1}{1+e^{-bx}} \tag{5.18}$$

where $x \in (-\infty, \infty)$ and $\text{sig}(x) \in (0,1)$. As shown in Figure 5.7(a), the parameter *b* controls the shape of the curve; larger *b* indicates faster convergence to 0 and 1. With the introduction of the sigmoid function, an *n*-dimensional decision space with unknown bounds can be transferred to a cube/hypercube bounded by $\mathbf{0} \in \mathbf{R}^n$ and $\mathbf{1} \in \mathbf{R}^n$. The DIRECT search will therefore be conducted within $(\mathbf{0}, \mathbf{1})$ range with the center point at $\mathbf{0.5} \in \mathbf{R}^n$. As illustrated in Figure 5.7(b), the search will have a better focus on the vicinity of zero vector by using larger *b*. The inverse sigmoid function is defined as,

$$x = \frac{1}{b}(\ln(\text{sig}(x)) - \ln(1-\text{sig}(x))) \tag{5.19}$$

In this research, *b* is set to be 1,000 based on our numerical tests so that it helps to properly facilitate sparse exploitation without the need to define hard bounds.



(a) (b)

Figure 5.7. Sigmoid and inverse sigmoid curves with b=1, 3, 5 and 15.

Finally, for an *h*-objective optimization problem, the pseudo-code of the sparse multi-objective DIRECT is presented as follows.

Algorithm **sparse multi-objective DIRECT**

1: Transfer the search space to a unit cube/hypercube through sigmoid function
2: Evaluate *h*-objectives $\mathbf{F}(\mathbf{c}_1) = \{ f_1(\mathbf{c}_1), f_2(\mathbf{c}_1), \ldots, f_h(\mathbf{c}_1) \}$; set $\{\boldsymbol{\alpha}_{\min}\} = \varnothing$, k = 1, and $k_{\max}$ = *max no. of evaluations*
3: **while** $k < k_{\max}$
4: Identify the set **S** of potentially optimal rectangles using Equation (17)
5: **while** $\mathbf{S} \neq \varnothing$
6: Take $j \in \mathbf{S}$
7: Sample new points, evaluate **F** at the new points and divide the rectangle *j* following step *2)*
8: $k := k + \Delta k$
9: $\mathbf{S} := \mathbf{S} \setminus \{j\}$ (remove $\{j\}$ from set $\mathbf{S}$)
10: **end while**
11: **end while**
12: **for** i = 1 to *k*
13: **if** $R(\mathbf{F}(\mathbf{c}_i)) = 1$ $\{\boldsymbol{\alpha}_{\min}\} := \{\boldsymbol{\alpha}_{\min}\} \bigcup sig^{-1}(\mathbf{c}_i)$
14: **end for**
15: **return** $\{\boldsymbol{\alpha}_{\min}\}$

The returned vector set $\{\boldsymbol{\alpha}_{\min}\}$ is the solution set obtained from Equation (5.12). A posterior articulation can be performed by omitting l_0 norm $f_2 = \|\hat{\boldsymbol{\alpha}}\|_0$ and then finding damage identification result $\hat{\boldsymbol{\alpha}}$ in $\{\boldsymbol{\alpha}_{\min}\}$ with smaller residue $f_1 = \|\mathbf{S}\hat{\boldsymbol{\alpha}} - \Delta\mathbf{Y}\|_2$. In line 13, $R(\mathbf{F}(\mathbf{c}_i)) = 1$ indicates that the solution has a rank index of 1, i.e., the solution belongs to the Pareto surface. We have now outlined the approach to be used. Figure 5.8 depicts the overall structure of the approach adopted for damage identification where the dashed boxes are the steps that promote sparsity of the solution.

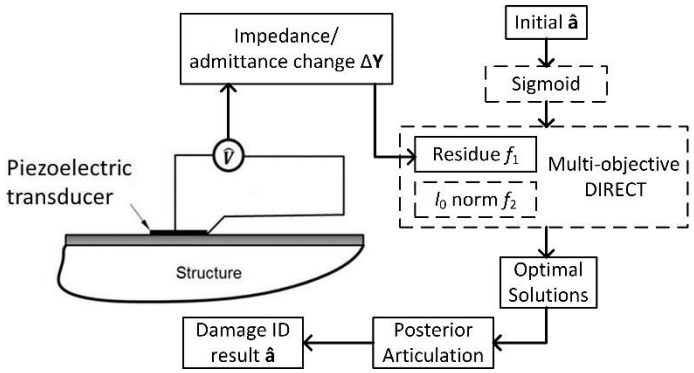


Figure 5.8. Damage identification using piezoelectric impedance/admittance measurement: overview.

5.4. Numerical case studies

In this section, we carry out comprehensive numerical case studies. We consider an aluminum cantilever plate with the following properties: length 0.561 m, width 0.01905 m, thickness 0.004763 m, density 2700 kg/m^3, and Young's modulus 68.9 GPa. The plate is subjected to the fixed-free boundary condition along the length-wise direction. As shown in Figure 5.9, A piezoelectric transducer is attached to this plate at location that is 0.18 m from the fixed end. The transducer has the following properties: length 0.015 m, width 0.01905 m, thickness 0.0014 m, Young's moduli $Y_{11} = 86$ GPa and $Y_{33} = 73$ GPa, density 9500 kg/m^3, piezoelectric constant -1.0288×10^9 V/m, and dielectric constant $\beta_{33} = 1.3832\times10^8$ m/F. The finite element model of the plate has 11,250 ($375\times15\times2$) 20-node hexahedron elements. It is further divided into 25 segments for damage identification indicating 25 possible damage locations. The segments are evenly divided along the length direction of the plate. In damage detection using admittance measurements, the admittance changes due to damage occurrence are most evident around the resonant peaks. Without loss of generality, we acquire the admittance change information around the plate's 14th (1893.58Hz: 1891.69Hz~1895.47Hz) and 21st (3704.05Hz: 3700.35Hz~3707.75Hz) natural frequencies. By keeping the linearly dependent rows of the sensitive matrix, a total of 21 useful measurements around the two natural frequencies are employed in the inverse analysis. As discussed in Section 5.1, the problem is under-determined. The linearization approximation

(Equation (5.5)) also introduces error. For both case studies presented below, the maximum number of function evaluation k_{max} is set as 100,000, which is sufficient the optimization problems to converge.

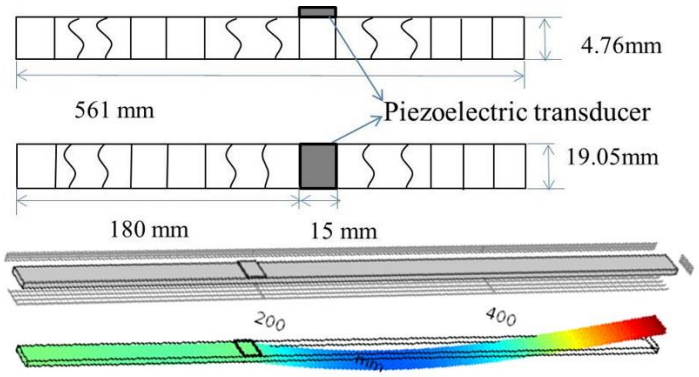


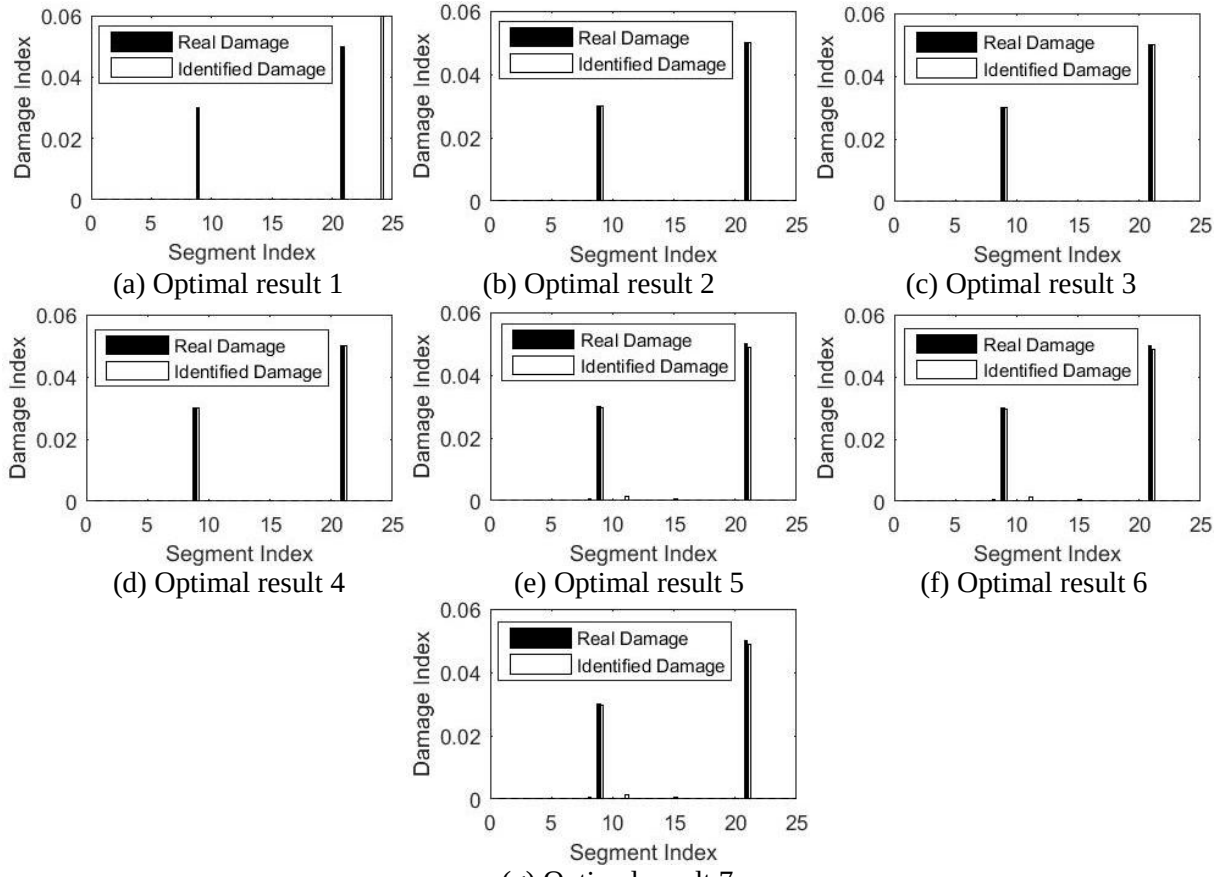
Figure 5.9. Coupled system setup for numerical case studies.

Table 5.1. Optimal results: numerical test case 1.

Optimal Results	Residue (f_1 in Equation (5.12))	# of Damage (f_2 in Equation (5.12))
1	1.8223e-07	1
2	1.1771e-10	2
3	1.1770e-10	3
4	1.1763e-10	4
5	1.1704e-10	5
6	1.1698e-10	6
7	1.1698e-10	7

We perform the first test with two randomly selected damage locations on the 9th and 21st segment with severities $\alpha_9 = 0.03$ (3% stiffness loss) and $\alpha_{21} = 0.05$ (5% stiffness loss) respectively. As shown in Table 5.1, multiple optimal solutions are produced by the sparse multi-objective DIRECT optimization. The solutions obtained exemplify the tradeoff between residue (Equation (5.12a)) and the number of damage locations (Equation (5.12b)). In other words, the solution with smaller residue may have more predicted damaged segments. As shown in Figures 5.10(b)-(g), optimal results 2 to 7 are relatively similar to each other in terms of identified results. In genetic algorithm description, they are considered to belong to the same niche. Consider optimal results 2 (Figure 5.10(b)) and 7 (Figure 5.10(g)) as example. The latter has two major damages (similar to optimal result 2) and six more negligible damages that reflect the error introduced by linear approximation and analytical modeling. Since the number of

damage is unknown in practice, we could consider the niches of optimal result 1 and 2 as possible candidates. However, as illustrated in Figure 5.11, optimal result 2 matches the input data better than optimal result 1 in terms of residue value. Thus, the niche of optimal result 2 is preferable, which indeed agrees with the true damage scenario.



(a) Optimal result 1 (b) Optimal result 2 (c) Optimal result 3

(d) Optimal result 4 (e) Optimal result 5 (f) Optimal result 6

(g) Optimal result 7

Figure 5.10. Numerical test case 1: real damage scenario compared with predicted damage scenarios.

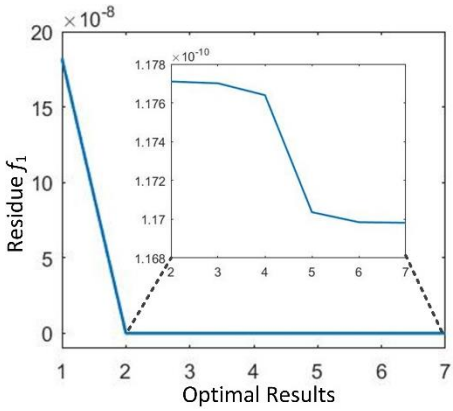
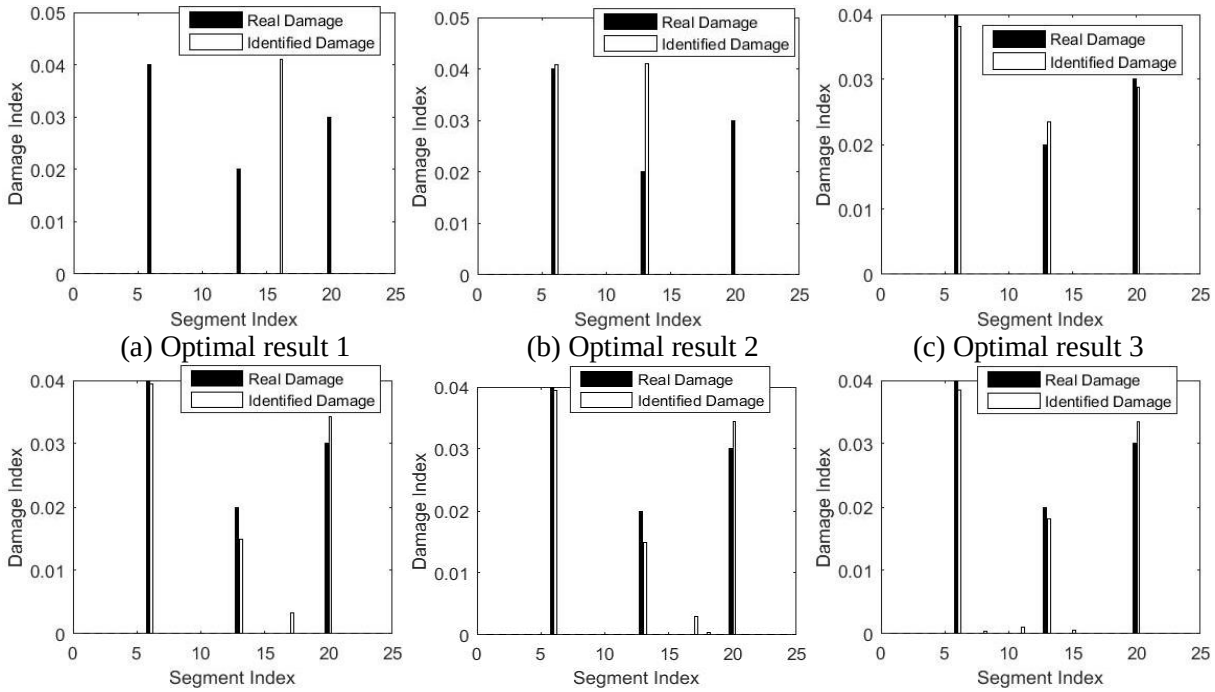


Figure 5.11. Numerical test case 1: Residue values of optimal results.

Table 5.2. Optimal results: numerical test case 2.

Optimal Results	Residue (f_1 in Equation (5.12))	# of Damage (f_2 in Equation (5.12))
1	8.2871e-07	1
2	2.4805e-08	2
3	7.0532e-13	3
4	2.0381e-13	4
5	1.9074e-13	5
6	2.9357e-14	6

Next we carry out the second case study with damage randomly selected at the 6^{th}, 13^{th}, and 20^{th} segments with severities of 0.04, 0.02 and 0.03 respectively. In this case, the number of damage is increased. Similarly, a set of optimal results consisting of the best solution for each possible number of damage are obtained and listed in Table 5.2. It is observed in Figure 5.12 that four out of six predictions successfully locate the damage and approximate the severities with excellent accuracy. Even though optimal results 2 and 3 are also plausible, by scrutinizing the residue values (Figure 5.13), we can find that solutions within the niche of optimal result 3 have smaller residue values, which indicate that they match with the input information better. For the proposed approach, when $k_{\max}$ is large enough, each optimal result obtained is the best result amongst its type in terms of the number of damage locations. Subsequently, even though such number is generally unknown during the inverse identification, the true damage scenario would be identified and the corresponding result will be included in the optimal set as long as the error is well-controlled and the problem is identifiable. In general, a posterior articulation can then be performed by selecting the results with smaller residue values. In certain situations when we know *a priori* that considerable modeling error and measurement noise are present, we can further look into results with relatively large residues. In any case, the proposed approach yields a small set of probable damage identification results with clear quantification of performance trade-offs (in terms of the number of damage locations and residue values). A final decision of damage identification can be reached by combining the piezoelectric admittance-based result with further inspection and/or additional monitoring schemes.



(a) Optimal result 1 (b) Optimal result 2 (c) Optimal result 3

(d) Optimal result 4 (e) Optimal result 5 (f) Optimal result 6

Figure 5.12. Numerical test case 2: real damage scenario compared with predicted damage scenarios.

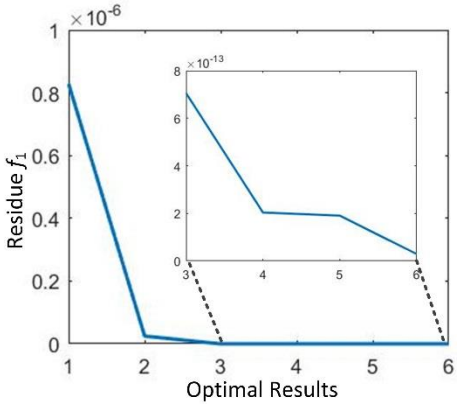
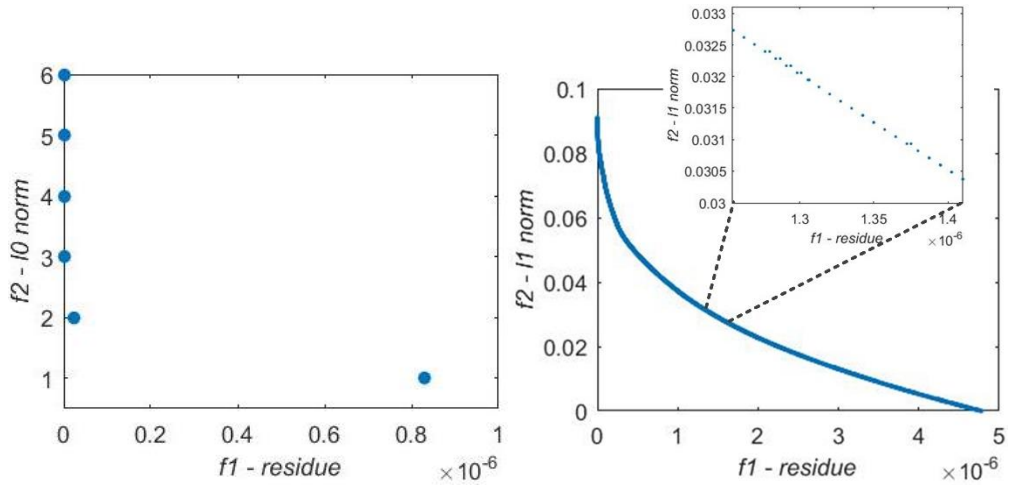


Figure 5.13. Numerical test case 2: Residue values of optimal results.

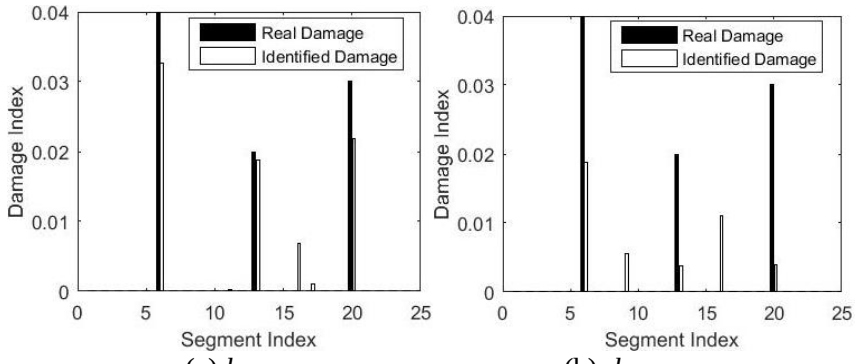
We further use the numerical study to elucidate the influence of the choice of sparsity regularizer. By changing $f_2 = \|\hat{\boldsymbol{\alpha}}\|_0$ in Equation (5.13b) to $f_2 = \|\hat{\boldsymbol{\alpha}}\|_1$ and applying sparse multi-objective sparse DIRECT, we conduct the comparison between the performance of l_0-norm and l_1-norm for the second numerical case study. When l_1-norm is employed, 1,344 optimal solutions are generated (Figure 5.14(b)) compared to 6 by l_0-norm (Figure 5.14(a)). Using l_1-norm poses challenge of performing a posterior articulation to start with. While for l_0-norm, even though it is not assured which candidate reflects the true damage state,

the results obtained provide guidance for further inspection and decision making, because the number of solutions is typically much smaller than the number of segments. Moreover, the quality of the solutions obtained by optimization using l_1-norm is worse than that of l_0-norm. As shown in Figure 5.15, the mean value of the optimal set of l_0-norm is more accurate and closer to the true damage state than l_1-norm as a whole. Additionally, the best solution in terms of the residue value, which corresponds to the point on the upper left corner in Figure 5.14, is 2.7343e-12 for l_1-norm, which is much larger than that of l_0-norm (2.9357e-14). As discussed in Section 5.2.1, l_1-norm serves as a convex approximation of l_0-norm so that the maximally sparse solutions are recovered with additional error.



(a) l_0-norm (b) l_1-norm

Figure 5.14. Optimal results of numerical test 2.



(a) l_0-norm (b): l_1-norm

Figure 5.15. Mean values of the predictions in the optimal sets.

5.5. Experimental validations

In this section, experimental cases using actual piezoelectric admittance measurements are carried out. The experimental setup, geometry measures and material parameters are consistent with those in the numerical cases analyzed in Section 4, as shown in Figure 5.16. To obtain the piezoelectric admittance, a resistor of $100\,\Omega$ is connected in series with the transducer to measure the voltage drop, which, together with the current in the circuit, yields the admittance information. A signal analyzer (Agilent 35670A) with a source channel is employed. The source channel is used to generate sinusoidal voltage sent to piezoelectric transducer (V_{in}), and the output voltage across the resistor is recorded (V_{out}). Hence, the admittance can be obtained,

$$Y_{\text{exp}} = \frac{I}{V_{\text{in}}} = \frac{V_{\text{out}}}{R_s V_{\text{in}}} \tag{5.20}$$

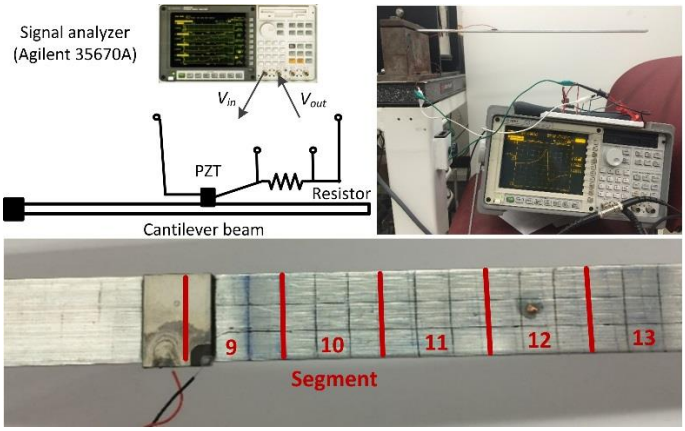
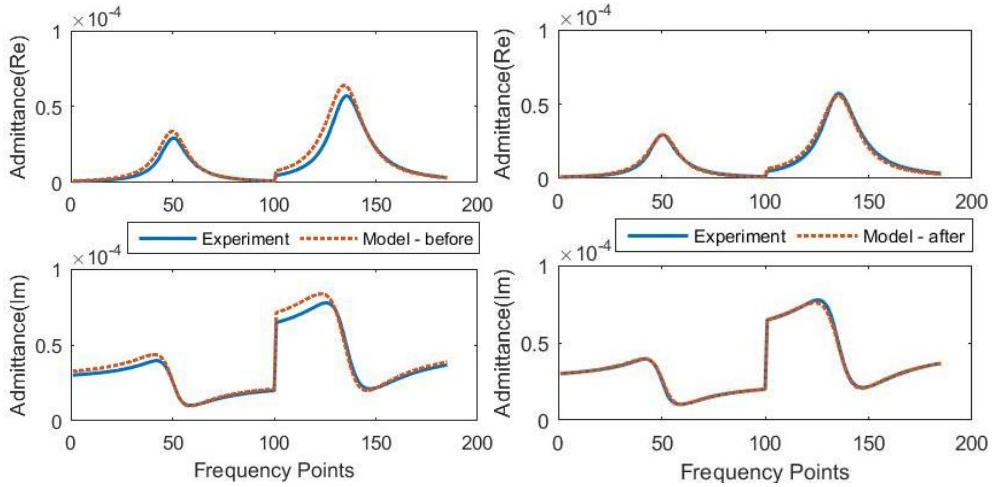


Figure 5.16 Experiment setup.

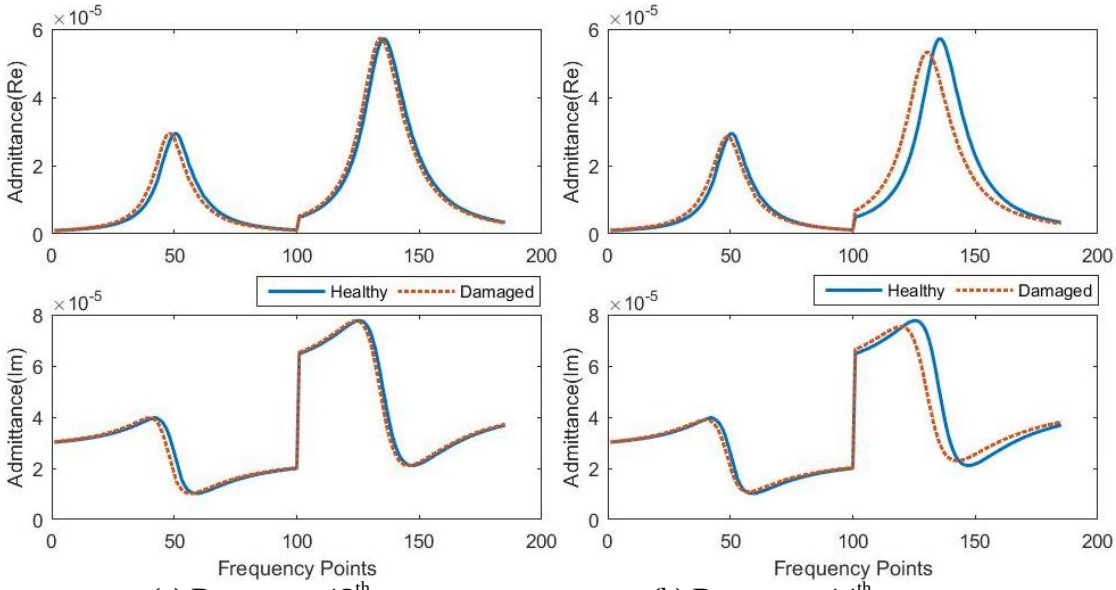
Meanwhile, we need to calibrate the finite element model of the healthy structure to match the experimental setup to minimize the modeling error usually induced by non-perfect boundary conditions. We update the boundary conditions through numerical optimization by identifying the stiffness values at the fixed end so the errors of the natural frequencies between the measurements and model predictions are minimized (Shuai et al, 2017). Comparisons between measured admittances and those generated by the

numerical model before and after model updating are illustrated in Figure 5.17. The model updating yields very good match with the healthy baseline.



(a) Before model updating (b) After model updating

Figure 5.17. Admittance measured by experiment vs. admittance generated by numerical model under healthy condition.



(a) Damage to 12th segment (b) Damage to 14th segment

Figure 5.18. Change of admittances (real and imaginary parts) caused by damage.

In order to minimize unwanted variations and uncertainties in the experiment, instead of disassembling and cutting the plate to reduce the local stiffness, we employ small added masses to emulate the damage occurrence. This can facilitate easy change of damage scenarios without altering the

experimental fixture. Practically, an added mass results in the change of admittance signatures equivalent to a local stiffness reduction. In the first experimental case, a 0.6 g mass is attached onto the plate at location corresponding to the 12th segment in the model, which causes admittance change equivalent to a 0.16% local stiffness loss (Figure 5.18(a)). In the second experiment, the same mass is attached onto the 14th segment of the plate which is equivalent to a 0.28% local stiffness loss (Figure 5.18(b)). Similar to numerical test cases, the admittances are measured around the 14th resonant frequency at 100 frequency points and the 21st resonant frequency at 85 frequency points. For both cases, the maximum number of function evaluation $k_{\max}$ is set as 100,000.

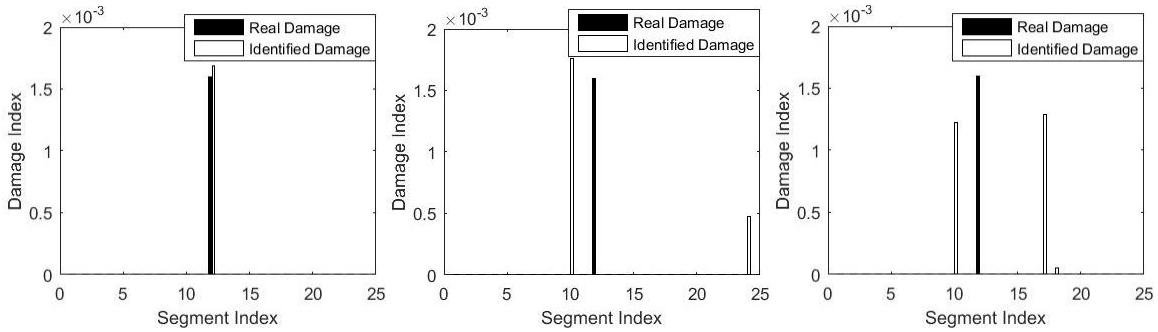
We first implement the new approach to tackle the case where the 12th segment is subjected to 0.16% stiffness loss. Three optimal results are acquired in Table 5.3 utilizing sparse multi-objective DIRECT. Unlike numerical case studies, there is no magnitude difference between the optimal results in terms of residue. While there is only linear approximation error in numerical case studies, model error and measurement error are inevitable in experiments, which in turn blur the boundary of solutions that reflect the admittance shift caused by damage or by errors. Thus, we consider each to be a candidate. As illustrated in Figure 5.19(a), optimal result 1 in our optimal set has the best accuracy. Nevertheless, given no prior knowledge of the number of damaged segments, optimal results 2 and 3 are also likely and they explain the as-measured data even better. Ideally, if the experimental setup and the numerical model have complete agreement, the new approach will converge to a single solution that matches the true damage scenario perfectly. But in practice, the convergent solution could be different because of error, and it is computationally intensive to merely minimize the residue without sparse relaxation. However, by using the proposed approach, not only the computational burden can be alleviated, but also, a small set of possible solutions containing the ones that are in close proximity of the true damage scenario are collected. This provides the foundation for subsequent inspections.

To better elaborate the significance of the proposed approach, Figure 5.20 provides an example of how decision space is sampled by the multi-objective DIRECT algorithm in both the sigmoid space and

the Euclidean space. Only 3 dimensions out of 25, which correspond to the 11th, 12th and 13th segment, are selected for visualization. It is shown that the "damaged" dimension 12 is intensively sampled while the samples in dimensions of α_{11} and α_{13} mostly stay around zero in Euclidean space.

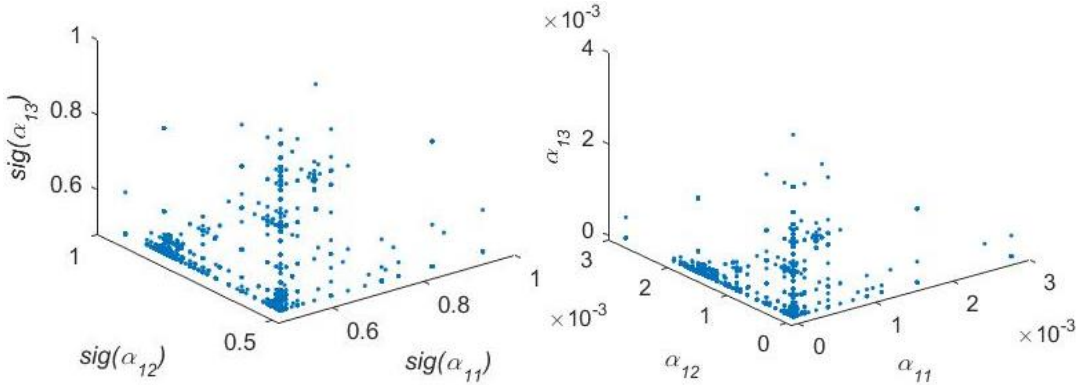
Table 5.3. Optimal results: experimental case 1

Optimal Results	Residue (f_1 in Equation (5.12))	# of Damage (f_2 in Equation (5.12))	Damage Scenarios
1	8.3545e-11	1	0.17% at 12nd
2	8.1254e-11	2	0.18% at 10th; 4.75e-2% at 23rd
3	8.1156e-11	3	0.12% at 10th; 0.13% at 17th; 4.94e-3% at 18th



(a) Optimal result 1 (b) Optimal result 2 (c) Optimal result 3

Figure 5.19. Experimental case 1: real damage scenario compared with predicted damage scenarios.



(a) Sigmoid space (b) Euclidean space

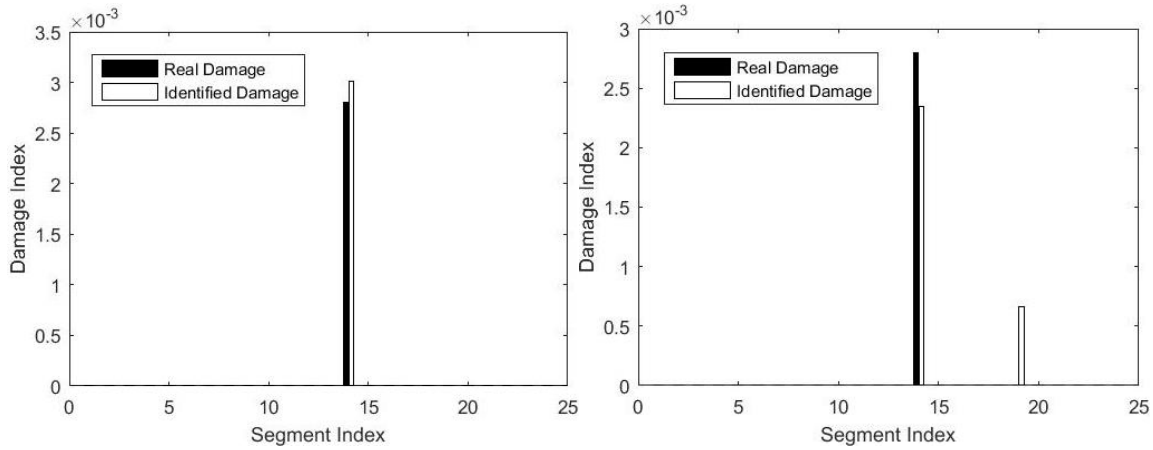
Figure 5.20. Experimental case 1: Multi-objective DIRECT Sampling.

Next, we perform the second experimental case study where the 14th segment is subjected to 0.28% stiffness loss. Table 5.4 lists four possible damage scenarios yielded by the proposed approach where the

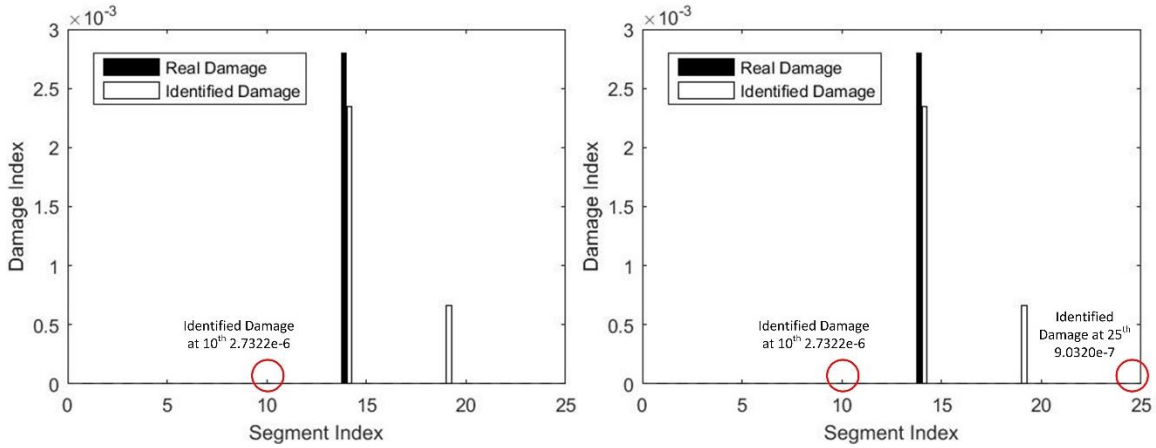
optimal result 1 agrees with the true damage scenario. As shown in Figure 5.21, optimal results 2 to 4 are very similar to each other and thus can be regarded as the niche of optimal result 2 altogether. Without knowing the number of damaged segments, the niche of optimal result 1 and optimal result 2 are both possible candidates. As shown in Figure 5.22, solutions with more damages interpret the data better. However, better interpretation does not necessarily mean better solution. In fact, the tradeoff between the residue and the l_0-norm can be regarded as the tradeoff between over-fitting and under-fitting. While errors are expected in SHM systems, the optimal set provided by the proposed approach actually unfolds the best options for further inspections. Take Figure 5.22(a) for example. If we want the result to reflect merely the main essence of the data, optimal result 1 on the lower right corner should be selected. On the contrary, optimal result 3 on the upper left corner should be chosen if we want the result to fit the data perfectly. This kind of explicit trade-offs yielded by the proposed approach can give clear guidance for further inspection, which matches with the typical procedure of inspection and maintenance in practice.

Table 5.4. Optimal results: experimental case 2.

Optimal Results	Residue (f_1 in Equation (5.12))	# of Damage (f_2 in Equation (5.12))	Damage Scenarios
1	2.4601e-09	1	0.3% at 14th
2	2.3634e-09	2	0.23% at 14th; 6.68e-2% at 18th
3	2.3634e-09	3	2.74e-4% at 10th; 0.23% at 14th; 6.65e-2% at 18th
4	2.3634e-09	4	2.74e-4% at 10th; 0.23% at 14th; 6.65e-2% at 18th; 9.14e-5% at 24th

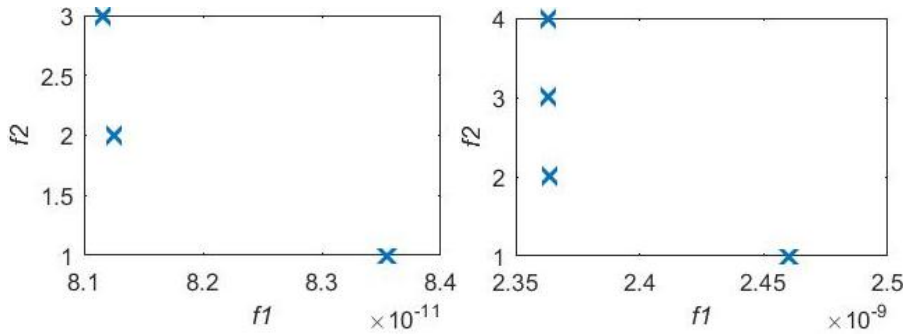


(a) Optimal result 1 (b) Optimal result 2



(c) Optimal result 3 (d) Optimal result 4

Figure 5.21. Experimental case 2: real damage scenario compared with predicted damage scenarios.



(a) Experiment 1 (b) Experiment 2

Figure 5.22. Optimal results of experimental cases.

5.6. Concluding remarks

This research presents an effective approach for structural damage identification utilizing piezoelectric admittance measurements. To address the fundamental challenge posed by the under-determined problem formulation that is rooted in the nature of high-frequency actuation/sensing, we cast the damage identification problem into a 2-objective optimization problem. The optimization problem is then tackled by a newly devised sparse multi-objective DIRECT algorithm. While the number of unknowns is very large in damage identification using high-frequency response measurements, in practical situations damage only affects a small number of segments. The proposed approach exploits the sparsity of the solution through l_0-norm minimization and sigmoid transformation. This not only reduces the number of solutions but also alleviates the computational burden for a global optimization. The

numerical tests and experiment validations demonstrate that the proposed algorithm is capable of obtaining a small set of high quality solutions that cover the true damage scenario. Instead of seeking for a deterministic solution which could be very different from the actual damage, this proposed approach utilizes SHM measurements to identify probable damage locations and severities that can be further examined/inspected. Such a procedure fits the practical SHM scheme, and thus can be used for complex structures theoretically. Our future works will focus on the effect of environmental conditions, such as temperature, external load, or the thickness of bonding layer, on the performance of the proposed approach.

Chapter 6. Leveraging Gaussian Process and Voting-Empowered Many-Objective Evaluation for Fault Identification

Using piezoelectric impedance/admittance sensing for structural health monitoring is promising, owing to the simplicity in circuitry design as well as the high-frequency interrogation capability. The actual identification of fault location and severity using impedance/admittance measurements, nevertheless, remains to be an extremely challenging task. A first-principle based structural model using finite element discretization requires high dimensionality to characterize the high-frequency response. As such, direct inversion using the sensitivity matrix usually yields an under-determined problem. Alternatively, the identification problem may be cast into an optimization framework in which fault parameters are identified through repeated forward finite element analysis which however is oftentimes computationally prohibitive. This paper presents an efficient data-assisted optimization approach for fault identification without using finite element model iteratively. We formulate a many-objective optimization problem to identify fault parameters, where response surfaces of impedance measurements are constructed through Gaussian process-based calibration. To balance between solution diversity and convergence, an epsilon dominance enabled many-objective simulated annealing algorithm is established. As multiple solutions are expected, a voting score calculation procedure is developed to further identify those solutions that yield better implications regarding structural health condition. The effectiveness of the proposed approach is demonstrated by systematic numerical and experimental case studies.

6.1 Introduction

The timely and accurate identification of faults in aerospace, mechanical, marine, and infrastructure systems has received significant recent attention. Different from traditional, offline non-destructive testing and evaluation (NDT&E) techniques, e.g., X-ray inspection, where the effectiveness is limited to the close vicinity of the sensors employed, online structural health monitoring is often facilitated through actuating and then sensing/measuring dynamic responses such as waves/vibrations that can propagate quite far (Farrar and Worden, 2006). This yields much larger coverage area and higher inspection

efficiency. The advent of many new transducer materials/devices and the advancement in microelectronics have resulted in rapid progresses in this area. On the other hand, bottlenecks and unique challenges exist. Structures are continuous media, and parameters characterizing structural faults, i.e., location and severity, are continuous variables as well. Hence, structural faults have infinitely many possible patterns/profiles with typically small characteristic lengths, which are further compounded by various uncertainties. Intuitively, the dynamic response data collected by the monitoring system must be in high-frequency range (i.e., with small wavelengths) so features of small-sized faults can be captured. The key issues thus are: 1) how to effectively generate high-frequency sensing data; and 2) how to efficiently and accurately identify fault location and severity from the data (Zhang et al, 2017).

Owing to their two-way electro-mechanical coupling, piezoelectric transducers are commonly used in structural health monitoring (Wang and Tang, 2008; Gao et al, 2018). One class of methods is ultrasonic propagating wave-based, where these transducers are used as actuators and sensors. The change of transient wave (e.g., Lamb wave) patterns, as waves propagate through fault site, can be used to infer fault occurrence. While these methods lead to high detection sensitivity due to the high-frequency nature, it is difficult to use transient responses to identify fault, especially to quantify the severity. The piezoelectric transducers have also been employed in the electrical impedance- or admittance-based methods where a piezoelectric transducer that is integrated (bonded/embedded) with the structure being monitored (Kim and Wang, 2014). In these methods, the piezoelectric transducer is driven by a sinusoidal voltage sweep over a certain frequency range, and the electrical response (i.e., the resulted current) is measured to extract the impedance/admittance information. Owing to the electro-mechanical coupling, the piezoelectric impedance/admittance is directly related to the mechanical impedance of the underlying structure. Thus, the change of piezoelectric impedance/admittance signature with respect to that under the healthy baseline state can be used as fault indicator. These methods have shown effectiveness for a variety of structural faults including crack, corrosion, debonding, joint degradation, etc (Park et al, 2008; Zhou and Zuo, 2012). The impedance/admittance can be measured in high-frequency range. A significant advantage is that in these methods the piezoelectric transducer serves as actuator and sensor

simultaneously and the circuitry design is extremely simple requiring essentially only a small resistor, which leads to implementation convenience.

A major hurdle remains. In theory, identifying directly the fault location and severity from stationary responses such as impedances/admittances is possible, as long as a credible first-principle model such as finite element model of the healthy baseline is available. A linearized sensitivity matrix can be derived that links the structural property changes to the changes of harmonic response magnitudes measured. In reality, such an inverse problem is usually severely under-determined. In order to characterize high-frequency impedance/admittance responses accurately, the finite element model must have high dimensionality. To pinpoint fault condition, we often divide the structure into a number of segments where the structural property in each segment is an unknown to be solved (because each segment is susceptible of fault occurrence). Therefore, the model has high dimensionality with a large number of unknowns. Meanwhile, structural faults manifest themselves in structural resonances and anti-resonances. As such, the effective measurements of impedance/admittance changes are limited (Shuai et al, 2017). One potential way to avoid the direct inversion is to convert the identification problem into an optimization problem, where possible property changes in all segments are treated as design parameters. These parameters are updated by minimizing the discrepancy between sensor measurements and model predictions through various optimization techniques in which only forward analyses are performed (Begambre and Laier, 2009; Perera et al, 2010; Cao et al, 2018a and 2018b). The necessary computational cost, however, could be very high. The forward optimization generally requires large number of iterations to converge, while a single run of high-dimensional finite element analysis can be very costly already.

Dynamic response calibration, as a faster alternative to exhaustive finite element analysis, has shown promising aspects in alleviating computational burden by emulating the full-scale finite element model responses. Traditional response surface methods applied for model updating use explicit functions to represent the relation between inputs and outputs. Least square-based techniques are then devised to refine parameters in the polynomial representation (Ren and Chen, 2010; Fang and Perera, 2011; Li and

Law, 2011; Chakrarborty and Sen, 2014). More recently, Gaussian process, also referred to as Kriging (Kennedy and O'Hagan, 2001; Rasmussen and Williams, 2006), has gained popularity due to its capability to simulate complicated process subjected to uncertainties. A Gaussian process model is not restricted to certain polynomial form thus allows highly flexible modeling in input-output relation based on statistical expectations and variances over functions. Gao et al (2013) used a Kriging surrogate model to calibrate frequency responses for crack tip location identification in cantilever plates. Yang et al (2017) proposed a similar calibration approach in frequency domain to detect the location and severity of fault in small structures. Wan and Ren (2015) suggested a residual-based Gaussian process model to characterize the relation between residual and updated parameters in frequency domain for finite element model updating. Jin and Jung (2016) formulated a sequential surrogate modeling scheme that constructs multiple response surfaces for finite element model updating. Balafas et al (2018) presented a Gaussian process model in wavelet domain that can infer damage through hypothesis testing. It is worth noting that all these dynamic response calibration methods are applied to natural frequency measurements. Since in practical situation only lower-order natural frequencies can be realistically measured, the case setups in these studies are relatively simple with low dimensionality and the design parameters are discrete with low dimensionality as well. In comparison, in impedance/admittance sensing, considerably more amount of measurements at many frequency points, can be acquired, and a high-dimensional structure is to be identified.

From the underlying physics standpoint, impedance/admittance sensing offers a new opportunity to identify fault parameters more accurately for more complex structures. While the response calibration technique appears to be promising in possibly avoiding iterative finite element analyses in an optimization framework, new issue arises. Although fault effects are reflected in impedance/admittance change at each frequency point theoretically, the actual impedance/admittance measurements respond to a fault condition differently at different frequencies. Therefore, in order to correctly identify fault conditions, one would need to examine the impedance/admittance changes at many frequency points. In other words, in order to take full advantage of the high-frequency impedance/admittance sensing, we need to formulate and then

solve efficiently an optimization problem to match response predictions with measurements at many frequency points. It should be noted that many-objective global optimization usually features more than three objectives, while multi-objective optimization refers to that with no more than three objectives. Although it would appear to be easier to resort to weighted summation to solve a single objective optimization (Gao et al, 2013; Wan and Ren, 2015; Yang et al, 2017), weighting selection among objectives is ad-hoc, and the result could easily converge to a meaningless outcome due to multiple local optima, measurement noise and uncertainties.

In this research, we develop a new methodology of fault identification using piezoelectric impedance/admittance sensing. To thoroughly elucidate the health status, a many-objective optimization is formulated to match parametric prediction with measurements at all frequency points of interest. Gaussian process regression is incorporated to construct the response surfaces, which not only significantly reduces computational cost but also yields continuous searching of fault parameters. Our goal in optimization is to find many solutions (owing to the under-determined nature of the problem) that are all optimal. In order to balance between solution convergence and diversity, we establish an ε-dominance enabled many-objective simulated annealing algorithm. Subsequently, inspired by concepts in social statistics, i.e., voting power and majority voting (Taylor and Pacelli, 2008), a voting score calculation framework is employed to evaluate quality of the solutions obtained. As a combination of many-objective optimization and voting score calculation, our proposed many-objective evaluation approach is able to distinguish the solutions that could accurately indicate the health condition of the structure and ultimately provide guidance for further examination.

The rest of this chapter is organized as follows. In Section 6.2, we establish the many-objective optimization formulation assisted by Gaussian process regression for piezoelectric impedance active sensing, where the ε-dominance enabled many-objective approach and the voting score calculation are presented in detail. In Section 6.3, the proposed method is evaluated through numerical case studies. Experimental validations are conducted in Section 6.4. Finally, concluding remarks are given in Section 6.5.

6.2 Approach formulation

6.2.1 Piezoelectric impedance/admittance active sensing

In piezoelectric impedance/admittance-based fault identification, a piezoelectric transducer circuit is attached to or embedded in a host structure. Harmonic excitation voltage, with frequency referred to as the excitation frequency or driving frequency, is supplied to actuate structural oscillation. The local structural oscillation in turn induces electrical response of the transducer due to electro-mechanical coupling. We can write the equations of motion of the coupled system in the finite element form as (Wang and Tang, 2010),

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} + \mathbf{K}_{12}Q = \mathbf{0} \tag{6.1a}$$

$$K_c Q + \mathbf{K}_{12}^{\mathrm{T}}\mathbf{q} = V_{\mathrm{in}} \tag{6.1b}$$

where **M**, **K** and **C** are the mass matrix, stiffness matrix and damping matrix, respectively, **q** is the structural displacement vector, $\mathbf{K}_{12}$ is the electro-mechanical coupling vector due to piezoelectric effect, K_c is the reciprocal of the capacitance of the piezoelectric transducer, Q is the electrical charge on the surface of the piezoelectric transducer, and V_{in} is the excitation voltage. Clearly in Equation (6.1), the impedance/admittance of the transducer is directly related to the impedance of the underlying structure and thus can be used as damage indicator. Under harmonic excitation, Equation (6.1) can be expressed in frequency domain. The admittance (reciprocal of impedance) of the piezoelectric transducer is then given as,

$$Y(\omega) = \frac{\dot{Q}}{V_{\mathrm{in}}} = \frac{\omega i}{K_c - \mathbf{K}_{12}^T(\mathbf{K} - \mathbf{M}\omega^2 + \mathbf{C}\omega i)^{-1}\mathbf{K}_{12}} \tag{6.2}$$

where ω is the excitation frequency and i is the imaginary unit. In discretized model-based fault identification, structural fault or damage is frequently assumed as local property change, e.g., local stiffness loss. We divide the host structure into n segments and use $\mathbf{k}_{hj}$ to represent the stiffness matrix of the j-th segment under healthy condition. The stiffness matrix of the structure when fault occurs can be

written as,

$$\mathbf{K}_d = \sum_{j=1}^{n} \mathbf{k}_{hj}(1-\alpha_j) \tag{6.3}$$

where $\alpha_j \in [0,1]$ is the fault index indicating the ratio of stiffness loss in the *j*-th segment. For example, if the *j*-th segment suffers from damage that leads to a 10% stiffness loss, then $\alpha_j = 0.1$, otherwise $\alpha_j = 0$. $\boldsymbol{\alpha} = [\alpha_1, \cdots, \alpha_n]^T$ is the fault index vector. As the piezoelectric transducer and the underlying structure form a coupled system, structural fault will be reflected by the admittance of the piezoelectric transducer,

$$Y_d(\omega, \boldsymbol{\alpha}) = \frac{\dot{Q}_d}{V_{\text{in}}} = \frac{\omega i}{K_c - \mathbf{K}_{12}^T(\mathbf{K}_d - \mathbf{M}\omega^2 + \mathbf{C}\omega i)^{-1}\mathbf{K}_{12}} \tag{6.4}$$

The measured admittance of the structure with fault can then be compared with the baseline admittance to elucidate the health condition. The change of admittance before and after fault occurrence can be written as a function of excitation frequency ω and damage index vector $\boldsymbol{\alpha}$,

$$\begin{aligned} \Delta Y(\omega, \boldsymbol{\alpha}) &= Y_d(\omega, \boldsymbol{\alpha}) - Y(\omega, \boldsymbol{\alpha} = \mathbf{0}) \\ &= \frac{\omega i(\mathbf{k}_h \mathbf{1} - \mathbf{M}\omega^2 + \mathbf{C}\omega i)(\mathbf{k}_h \boldsymbol{\alpha} - \mathbf{M}\omega^2 + \mathbf{C}\omega i)}{-\mathbf{K}_{12}^T \mathbf{k}_h} \end{aligned} \tag{6.5}$$

In Equation (6.5), $\mathbf{k}_h = [\mathbf{k}_{h1}, \cdots, \mathbf{k}_{hn}]$, which represents the stiffness sub-matrices of *n* segments when the structure is healthy. In impedance/admittance-based fault identification, as harmonic voltage excitation is supplied for active sensing, Equation (6.5) is used iteratively giving different read of $\Delta Y(\omega, \boldsymbol{\alpha})$ when the excitation frequency is swept within certain ranges that cover a number of structural resonances around which physical measurements are taken. In order to characterize high-frequency responses, the finite element model must have high dimensionality.

6.2.2 Data-assisted impedance response calibration

As indicated in Introduction, direct inverse analysis based on Equation (6.5) generally yields a severely under-determined problem. One possible solution is to perform repeated forward finite element analyses in the parametric space within an optimization framework to identify fault parameters. In order

to render such a procedure computationally tractable, in this sub-section we present a data-assisted meta-modelling approach through Gaussian process (GP) regression (Kennedy and O'Hagan, 2001; Rasmussen and Williams, 2006; Stein, 2012). Essentially, we aim at rapidly constructing the response surfaces in the parametric space through emulations using experimental and/or numerical simulation data.

Gaussian process (GP) regression is an interpolation approach by which various spatial and temporal problems can be modeled (Xia and Tang, 2013; Wang et al, 2017). For impedance-based active sensing, the observed output can be symbolized and denoted as $\Delta Y(\mathbf{x}) = f(\mathbf{x}) + \varphi$, where $f(\mathbf{x})$ is the output of the numerical model, $\mathbf{x}$ is the input vector, and φ is the model discrepancy. The additive error φ is assumed to follow an independent, identically distributed Gaussian distribution $\varphi \sim N(0, \sigma_n^2)$. A function $\phi(\mathbf{x})$ can be introduced to map the input $\mathbf{x}$ to $f(\mathbf{x})$ as,

$$f(\mathbf{x}) = \phi(\mathbf{x})^T \mathbf{w} \tag{6.6}$$

where $\mathbf{w}$ is a vector of unknown parameters. The probability density of the set of training samples $(\Delta\mathbf{Y}, \mathbf{X})$ given $\mathbf{w}$ can then be obtained,

$$\begin{aligned} &p(\Delta\mathbf{Y}|\mathbf{X}, \mathbf{w}) = \prod_{i=1}^{n} p(\Delta Y_i | \mathbf{x}_i, \mathbf{w}) = \\ &\prod_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma_n} \exp(-\frac{(\Delta Y_i - \phi(\mathbf{x}_i)^T \mathbf{w})^2}{2\sigma_n^2}) \sim N(\phi(\mathbf{x})^T \mathbf{w}, \sigma_n^2 \mathbf{I}) \end{aligned} \tag{6.7}$$

The training samples can be acquired either experimentally or from a credible finite element model. Now we assume a multivariate Gaussian prior over the parameters $\mathbf{w} \sim N(\mathbf{0}, \mathbf{\Sigma}_p)$ with zero mean and certain covariance. We can obtain the posterior probability density of $\mathbf{w}$ through Bayes' theorem,

$$p(\mathbf{w}|\mathbf{X}, \Delta\mathbf{Y}) = \frac{p(\Delta\mathbf{Y}|\mathbf{X}, \mathbf{w}) p(\mathbf{w})}{p(\Delta\mathbf{Y}|\mathbf{X})} \sim (\sigma_n^{-2} A^{-1} \phi(\mathbf{x}) \Delta\mathbf{Y}, A^{-1}) \tag{6.8}$$

where $A = \sigma_n^{-2} \phi(\mathbf{x}) \phi(\mathbf{x})^T + \mathbf{\Sigma}_p^{-1}$. Finally, by averaging over all possible parameter values, the predictive distribution of f_* given a test input vector $\mathbf{x}_*$ also follows Gaussian distribution,

$$p(f_*|\mathbf{x}_*,\mathbf{X},\Delta\mathbf{Y}) = \int p(f_*|\mathbf{x}_*,\mathbf{w})p(\mathbf{w}|\mathbf{X},\Delta\mathbf{Y})d\mathbf{w} \sim N(\sigma_n^{-2}\phi(\mathbf{x}_*)^T A^{-1}\phi(\mathbf{x})\Delta\mathbf{Y}, \phi(\mathbf{x}_*)^T A^{-1}\phi(\mathbf{x}_*)) \quad (6.9)$$

Therefore, any finite number of outputs f_* given multiple test inputs $\mathbf{x}_*$ have a joint Gaussian distribution. To define such distribution over the stochastic process $f(\mathbf{x})$, a Gaussian process regression model is developed,

$$f(\mathbf{x}) \sim GP(m(\mathbf{x}), k(\mathbf{x},\mathbf{x}')) \quad (6.10)$$

Equation (6.10) is fully specified by its mean function $m(\mathbf{x})$ and covariance function or kernel $k(\mathbf{x},\mathbf{x}')$ where $\mathbf{x}$ and $\mathbf{x}'$ are in either the training or the test sets. For prior $\mathbf{w} \sim N(\mathbf{0}, \mathbf{\Sigma}_p)$, the mean and covariance functions that determine the smoothness and variability are written as,

$$m(\mathbf{x}) = \mathrm{E}[f] = \phi(\mathbf{x})^T E[\mathbf{w}] = \mathbf{0} \quad (6.11)$$

$$k(\mathbf{x},\mathbf{x}') = \mathrm{E}[(f - m)(f' - m')] = \phi(\mathbf{x})^T \mathbf{\Sigma}_p \phi(\mathbf{x}') \quad (6.12)$$

The joint distribution of observation $\Delta\mathbf{Y}$ and unknown output $\mathbf{f}_*$ given training input set $\mathbf{X}$ and test input set $\mathbf{X}_*$ is then,

$$\begin{bmatrix} \Delta\mathbf{Y} \\ \mathbf{f}_* \end{bmatrix} \sim N\left(\mathbf{0}, \begin{bmatrix} K(\mathbf{X},\mathbf{X}) + \sigma_n^2\mathbf{I} & K(\mathbf{X},\mathbf{X}_*) \\ K(\mathbf{X}_*,\mathbf{X}) & K(\mathbf{X}_*,\mathbf{X}_*) \end{bmatrix}\right) \quad (6.13)$$

$K(\mathbf{X},\mathbf{X}_*)$ denotes the $n \times n_*$ matrix of kernels evaluated at all pairs of training and test points through $k(\mathbf{x},\mathbf{x}_*)$ for n training samples and n_* test inputs. Finally, the key predictive distribution of Gaussian process regression, i.e., the conditional distribution of $\mathbf{f}_*$, is expressed as

$$p(\mathbf{f}_*|\mathbf{X}_*,\mathbf{X},\Delta\mathbf{Y}) \sim N(K(\mathbf{X}_*,\mathbf{X})[K(\mathbf{X},\mathbf{X}) + \sigma_n^2\mathbf{I}]^{-1}\Delta\mathbf{Y}, K(\mathbf{X}_*,\mathbf{X}_*) - K(\mathbf{X}_*,\mathbf{X})[K(\mathbf{X},\mathbf{X}) + \sigma_n^2\mathbf{I}]^{-1}K(\mathbf{X},\mathbf{X}_*)) \quad (6.14)$$

which is the function-space view of Equation (6.9). In this research, the input vector is given as $\mathbf{x} = [\omega, \boldsymbol{\alpha}] \Leftrightarrow [\omega, \alpha_L, \alpha_S]$, where ω is the excitation frequency, and $\boldsymbol{\alpha}$ is the fault index vector. The vector $\boldsymbol{\alpha}$ can be further expressed as $[\alpha_L, \alpha_S]$ for single fault cases, where α_L is the location and α_S is the severity. For example, if a structure is divided into 6 segments and the 3rd segment is subjected to 5%

damage (5% stiffness loss), then $\boldsymbol{\alpha}=[0, 0, 0.05, 0, 0, 0]$ or $\boldsymbol{\alpha} \Leftrightarrow [\alpha_L, \alpha_S] = [3, 0.05]$. For each given ω_j ($j = 1, 2, ..., l$) where l is the number of frequency points swept during inspection, if m observations or training data in Gaussian process regression can be obtained as $\mathbf{D}_j = \left\{ (\Delta Y_{ji}, \alpha_{Li}, \alpha_{Si}) \middle| i = 1, 2, ..., m \right\}$, we can then have l calibrations trained by $\{\mathbf{D}_1, \mathbf{D}_2, ..., \mathbf{D}_l\}$ with a Gaussian process regression model $f(\boldsymbol{\alpha}) \sim GP(\mathbf{0}, k(\boldsymbol{\alpha}, \boldsymbol{\alpha}'))$. One of the most widely-adopted kernel functions is the squared exponential function (Rasmussen and Williams, 2006),

$$k(\boldsymbol{\alpha}, \boldsymbol{\alpha}') = \theta_1 \exp\left(\frac{\left| \boldsymbol{\alpha} - \boldsymbol{\alpha}' \right|^2}{\theta_2} \right) \tag{6.15}$$

which is efficient toward cases where the training data is of the same type but in different dimensions. For inputs that have more than one type of feature, such as $[\alpha_L, \alpha_S]$ characterizing location and severity that are different in nature, a well-accepted way to build a kernel is to multiply kernels together (Duvenaud, 2014). In this research, we adopt the product of two squared exponential functions as kernel,

$$k(\boldsymbol{\alpha}, \boldsymbol{\alpha}') = \theta_1 \exp\left(\frac{\left| \boldsymbol{\alpha} - \boldsymbol{\alpha}' \right|^2}{\theta_2} \right) \bullet \theta_3 \exp\left(\frac{\left| \boldsymbol{\alpha} - \boldsymbol{\alpha}' \right|^2}{\theta_4} \right) \tag{6.16}$$

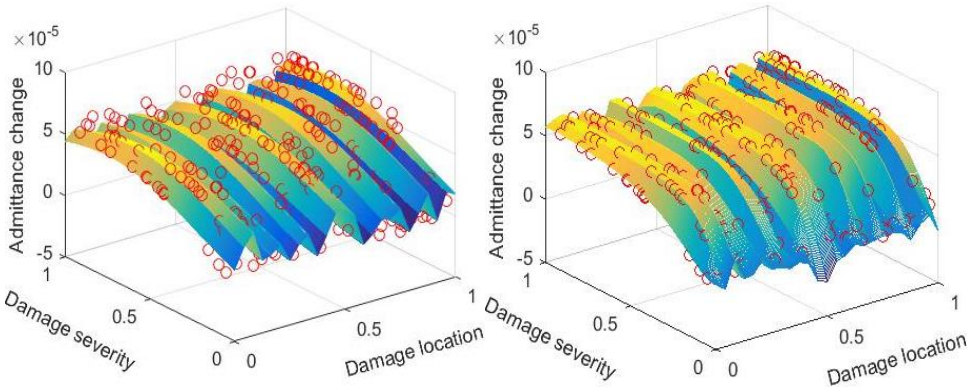
The hyper-parameters $\boldsymbol{\theta}$ used in kernel are trained by maximizing the marginal likelihood $p(\Delta \mathbf{Y} | \mathbf{X})$, or the log marginal likelihood w.r.t. $\boldsymbol{\theta}$ and σ_n,

$$\begin{aligned} \log p(\Delta \mathbf{Y} | \boldsymbol{\alpha}) = & -\frac{1}{2} \Delta \mathbf{Y}^T (K(\boldsymbol{\alpha}, \boldsymbol{\alpha}) + \sigma_n^2 \mathbf{I})^{-1} \Delta \mathbf{Y} \\ & -\frac{1}{2} \log \left| K(\boldsymbol{\alpha}, \boldsymbol{\alpha}) + \sigma_n^2 \mathbf{I} \right| - \frac{n}{2} \log 2\pi \end{aligned} \tag{6.17}$$

The parameters are then evaluated using Markov chain Monte Carlo method (Neal, 2011) in this study.

Compared to single squared exponential kernel (Equation (6.15)), the product of squared exponential kernels (Equation (6.16)) can better represent the training samples in impedance-based fault identification. As shown in Figure 1, admittance changes are calibrated using single squared exponential function as kernel and product of squared exponential functions as kernel, respectively, given 270 training

data. The calibration surface is the mean value of the predictive distribution acquired using Equation (6.14).



(a) (b)

Figure 6.1 Admittance change calibrations (a) single squared exponential kernel (b) product of squared exponential kernels (○ : training sample)

For l frequencies ω_j ($j = 1, 2, ..., l$) swept by a piezoelectric transducer in active sensing, if l sets of training data $\mathbf{D}_j$ are available either by experiment or from a finite element model, l calibration surfaces similar to Figure 6.1(b) can be developed,

$$\begin{matrix} \Delta Y_1^{(c)}(\alpha_L, \alpha_S)|\mathbf{D}_1 \\ \vdots \\ \Delta Y_l^{(c)}(\alpha_L, \alpha_S)|\mathbf{D}_l \end{matrix} \tag{6.18}$$

where $\Delta Y_j^{(c)}$ represents the output of the reconstructed surface for any input (α_L, α_S) under excitation frequency ω_j.

The proposed method therefore utilizes the regression models to reproduce responses by given different arguments of the response surfaces (health condition of the structure), where the analytical sensitivity matrix to correlate variables with the response is not involved. By minimizing the discrepancy between the predictions made by reconstructed surfaces and the actual measurements, the fault identification problem is essentially cast into an optimization problem. The impedance/admittance changes measured physically under the same l excitation frequencies $\Delta Y_l^{(m)}$ are used to form l objective

functions,

$$\begin{aligned} \min J_1 &= \left|\Delta Y_1^{(c)}(\alpha_L, \alpha_S) - \Delta Y_1^{(m)}\right| \\ &\vdots \\ \min J_l &= \left|\Delta Y_l^{(c)}(\alpha_L, \alpha_S) - \Delta Y_l^{(m)}\right| \end{aligned} \tag{6.19}$$

where α_S and α_L are the design variables of the optimization problem. Consider the case where only one objective function J_1 is employed. Minimizing merely J_1 will possibly yield a large number of wrong solutions because it is an under-determined problem with only one measurement subjected to error. Clearly, more information regarding the health condition should be taken into consideration by employing more objective functions. This showcases the underlying reason we formulate a many-objective optimization problem. We aim to find the "overlapping consensus" among the available, many objective functions. It is, however, computationally challenging to solve such an optimization problem.

6.2.3 Voting-empowered many-objective evaluation

Many-objective optimization (MaOO) problems are defined as those with four or more objectives (Deb and Jain, 2014) where the results cannot be directly visualized through graphical means. In comparison, multi-objective optimization problems have two or three objective functions. To illustrate the difficulties associated with solving many-objective optimization problems, we first introduce the Pareto optimality based multi-objective optimization, which has seen extensive research efforts (Zitzler, 1999; Deb et al, 2002; Laumanns et al, 2002; Zhang and Li, 2007; Cao et al, 2017). For multi-objective optimization, the Pareto optimality is defined in a broader sense that no other solution is superior to the Pareto optimal solutions when all objectives are considered. Following this, a general Pareto-based MaOO problem where *n* objectives are minimized simultaneously is specified as

$$\text{Minimize} \quad \boldsymbol{f}(\mathbf{x}) = \{ f_1(\mathbf{x}), \ldots, f_n(\mathbf{x}) \} \tag{6.20}$$

where $\mathbf{x}$ is the decision vector and $\boldsymbol{f}$ is the objective vector. When two sets of decision vectors are compared, the concept of dominance is involved. Assuming **a** and **b** are two decision vectors, the concept of Pareto optimality can be defined as follows: **a** dominate **b** if:

$$\forall i = \{1,2,...,n\}: f_i(\mathbf{a}) \leq f_i(\mathbf{b}) \tag{6.21}$$

and

$$\exists\, j = \{1,2,...,n\}: f_j(\mathbf{a}) < f_j(\mathbf{b}) \tag{6.22}$$

Any objective function vector, which is neither dominated by any objective function vector in the Pareto optimal set nor dominating any of them, is said to be non-dominated with respect to that Pareto optimal set. The solution that corresponds to the objective function vector is then a member of Pareto optimal set.

In comparison with multi-objective optimization, many-objective evaluation needs to tackle two major additional difficulties (Deb and Jain, 2014; Li et al, 2015):

1) Almost all solutions generated are non-dominated to one another. As the number of objectives increases, even a mediocre solution could be Pareto optimal because it may have small advantages over other solutions in at least one objective, even though the differences are trivial. Consequently, most Pareto optimality-based multi-objective optimization algorithms become inefficient and out of focus when dealing with many objectives. The solution set yielded may be arbitrarily large.

2) It is hard to maintain good diversity among the solution set in high dimensional space. Generally, it is computationally expensive to evaluate diversity for many objectives. Moreover, the conflict between convergence and diversity is aggravated in high dimension. Therefore, attempts to maintain diversity may hinder the numerical procedure from converging to the optimal solutions.

The difficulties can be alleviated by using a special domination principle that will adaptively discretize the Pareto optimal set and find a well-distributed set of solutions. A good choice to tackle the above-mentioned difficulties is the ε-dominance principle (Laumanns et al, 2002), which alters and discretizes the objective space into boxes defined by the power of $(1+\varepsilon)$,

$$\left\lfloor \frac{\log f_i}{\log(1+\varepsilon)} \right\rfloor \tag{6.23}$$

Equation (6.23) projects each objective function vector uniquely to one box, which can neutralize

trivial improvements in any objectives. One example is shown in Figure 6.2, one Pareto optimal solution (1.5, 3.5) in the original objective space is eliminated in the ε-Pareto optimal set because it is merely better in one objective but a lot worse in the other objective compared to solution (1.6, 2.5). And by keeping one solution per box, a bounded size solution set with good diversity could be obtained. The aforementioned difficulties can thus be addressed by the ε-dominance transformation. Accordingly, the dominance relation based on ε-dominance is given in Table 6.1 where the box operator refers to Equation (6.23) and $\prec$ is used to denote dominance relation between decision vectors.

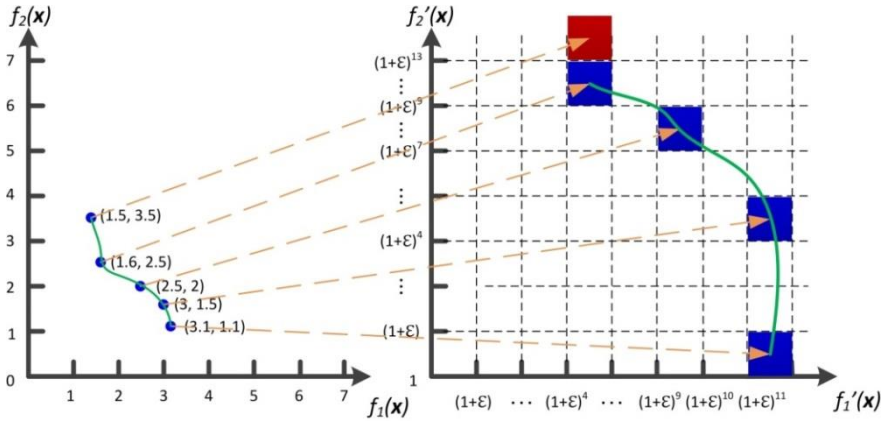


Figure 6.2 From Pareto optimal to ε-Pareto optimal

Table 6.1 ε-dominance relations

Relation	Symbol	Interpretation in ε-objective space
box(**a**) dominates *box*(**b**)	$box(\mathbf{a}) \prec box(\mathbf{b})$	*box*(**a**) is not worse than *box*(**b**) in all objectives and better in at least one objective
box(**b**) dominates *box*(**a**)	$box(\mathbf{b}) \prec box(\mathbf{a})$	*box*(**b**) is not worse than *box*(**a**) in all objectives and better in at least one objective
Non-ε-dominant to each other	$box(\mathbf{b}) \cong box(\mathbf{a})$	*box*(**a**) is worse than *box*(**b**) in some objectives but better in some other objectives
Same box	$box(\mathbf{a}) = box(\mathbf{b})$	*box*(**a**) equals *box*(**b**)

We incorporate the ε-dominance technique into a previously developed Multi-objective Simulated Annealing algorithm (Cao et al, 2017), hereafter referred to as ε-MOSA/R. The pseudo-code of ε-MOSA/R is provided below.

Algorithm ε-MOSA/R

Set *Tmax*, *Tmin*, # of iterations per temperature *iter*, cooling rate α, $k = 0$

Initialize the *Archive* (ε -Pareto front)
Current solution = randomly chosen from *Archive*
While (*T* > *Tmin*)
For 1 : *iter*
Generate a *new solution* vector in the neighborhood of *current solution* vector
If *new solution* falls into the same ε -box as any solutions in the *Archive*
If *new solution* dominates *k* (*k* >= 1) solutions in the *Archive*
Update
Else
Action
End if
Else if *new solution* ε -dominates *k* (*k* >= 1) solutions in the *Archive*
Update
Else if *new solution* ε -dominated by *k* (*k* >= 1) solutions in the *Archive*
Action
Else if *new solution* and *Archive* are non-ε -dominant to each other
Update
End if
End for
$k = k+1$
$T = (\alpha^{k})*Tmax$
End While

Algorithm Update

Remove all *k* dominated solutions from the *Archive*
Add *new solution* to the *Archive*
Set *new solution* as *current solution*

Algorithm Action

If *new solution* and *Archive* are non-dominant to each other
Set *new solution* as *current solution*
Else
If *new solution* dominated by *current solution*
Re-seed
Else
Simulated Annealing
End If
End if

Algorithm Re-seed

new solution is dominated by *k* (*k* >= 1) solutions in the *Archive*
Set *selected solution* as the *i*-th solution $i = \arg\min_{i=1,2,...,k}(\Delta dom_{i,new})$
If $\frac{1}{1+\exp(-\Delta dom_{selected,new} / \max(T,1))} > \text{rand}(0,1)*$
Set *selected solution* as *current solution*
Else
Simulated Annealing
End if

***** rand(0,1) generates a random number between 0 to1

Algorithm Simulated Annealing

$$\Delta dom_{avg} = \frac{\sum_{i=1}^{k} \Delta dom_{i,new}}{k}$$

If $\frac{1}{1+\exp(\Delta dom_{avg} / T)} > \text{rand}(0,1)$

Set *new solution* as *current solution*

End if

In ε-MOSA/R, we use ε-dominance relation as well as the regular dominance relation to compare the new solution, the current solution and Archive. Algorithm Update renews the Archive when a better solution in ε-dominance sense is found and meanwhile assures that only one solution is maintained per ε-box. As Algorithm Re-seed and Algorithm Simulated Annealing are embedded, Algorithm Action takes place when a deteriorated solution is sampled. Instead of abandoning the solution directly, probability relaxations are devised so that either the deteriorated solution is accepted with a certain probability to escape local optima (Simulated Annealing) or the search direction is swerved towards known search space with good solutions for better efficiency (Re-seed). The concept of the amount of domination is used in computing the acceptance probability in Re-seed and Simulated Annealing (Bandyopadhyay et al., 2008). Given two solutions **a** and **b**, the amount of domination is defined as

$$\Delta dom_{\mathbf{a},\mathbf{b}} = \prod_{i=1, f_i(\mathbf{a}) \neq f_i(\mathbf{b})}^{l} (|f_i(\mathbf{a}) - f_i(\mathbf{b})| / R_i) \tag{6.20}$$

where l is the number of objectives and R_i is the range of the i-th objective for normalization. In this research, for all case studies to converge, the total number of iterations of ε-MOSA/R is set as 100,000, *Tmax* is 100, *Tmin* is 10^{-4}, and the cooling rate α is set as 0.8.

Ideally, if the calibration surfaces are perfect without error, using more objectives (i.e., incorporating more measurements) naturally yields solution sets of better accuracy. If sufficient response surfaces are used, the solution set should contain only one solution that matches perfectly the fault scenario. However, owing to modeling and calibration errors, utilizing more objective functions does not necessarily associate with better performance. As seen in Equation (19), l objective functions can be

formulated under l excitation frequencies. While ε-MOSA/R introduced could cope with such a many-objective optimization problem, the solution size would increase nonlinearly with l (Duro et al, 2014). Therefore, it could be even harder for a greater number of calibration surfaces to reach the "overlapping consensus" in determining the structural damage. Although using some subsets of available calibrated surfaces can uncover a small set of trustworthy solutions for further analysis, using other subsets may return a large number of erroneous results. Given the difficulties, guiding the algorithm to only a few optimal solutions or making an objective decision becomes critical in many-objective optimization.

In this research, inspired by social statistics (Taylor and Pacelli, 2008), we introduce a novel voting score calculation procedure based on the concepts of majority voting and voting power to evaluate the quality of the solutions generated using different sets of response surfaces as objective functions. As not all those l objective functions are essential or equally important, to reduce variance, N ($N \le l$) functions are randomly selected from the set as objectives of the many-objective optimization problem denoted as $\mathbf{J}$, which can be deemed as input of the many-objective algorithm,

$$\mathbf{A} = \varepsilon\text{-MOSA/R}(\mathbf{J}) \tag{6.21}$$

where $\mathbf{A} = \{\boldsymbol{\alpha}_a, \boldsymbol{\alpha}_b, \boldsymbol{\alpha}_c, ...\}$ represents the set of Pareto optimal solutions obtained after one many-objective optimization.

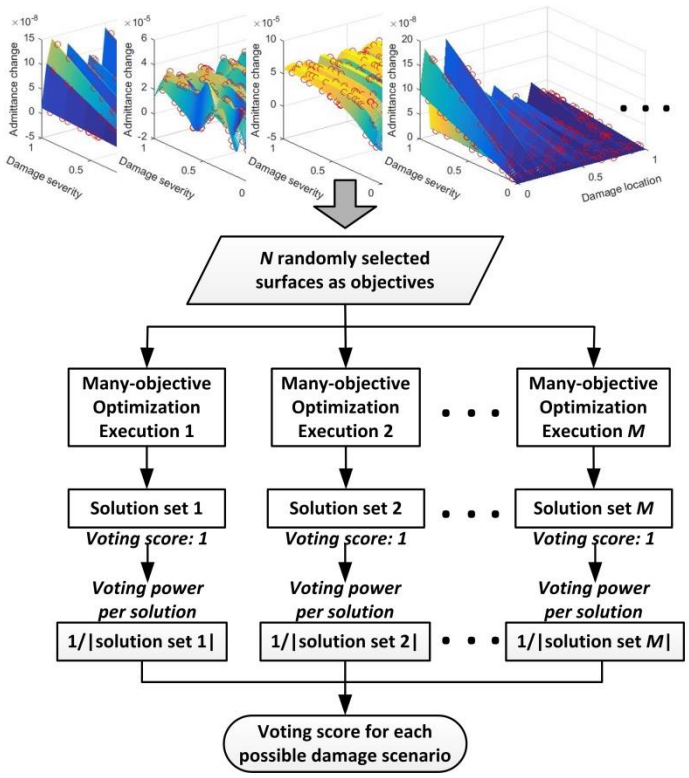


Figure 6.3 Voting score calculation for multi-objective evaluation

As shown in Figure 6.3, the many-objective optimization proposed is carried out *M* times for *N* randomly selected response surfaces as objective functions different for each execution. Hence, we have,

$$\{\mathbf{A}_1, \mathbf{A}_2, ..., \mathbf{A}_M\} = \varepsilon\text{-MOSA/R}(\{\mathbf{J}_1, \mathbf{J}_2, ..., \mathbf{J}_M\}) \tag{6.22}$$

We use $\mathbf{A}_i$ to represent the solution set of the *i*-th execution of the optimization given objective function set $\mathbf{J}_i$. The voting score for a specific solution is calculated as,

$$\text{vs}(\boldsymbol{\alpha}_a) = \sum_{i=1}^{M} \left| \boldsymbol{\alpha}_a \bigcap \mathbf{A}_i \right| / \left| \mathbf{A}_i \right| \tag{6.23}$$

where $\left| \boldsymbol{\alpha}_a \bigcap \mathbf{A}_i \right|$ equals to 1 if $\mathbf{A}_i$ contains $\boldsymbol{\alpha}_a$. For example, if $\boldsymbol{\alpha}_a$ appears in optimal solution set $\mathbf{A}_1$, $\mathbf{A}_3$ and $\mathbf{A}_4$, then $\text{vs}(\boldsymbol{\alpha}_a) = 1/\left|\mathbf{A}_1\right| + 1/\left|\mathbf{A}_3\right| + 1/\left|\mathbf{A}_4\right|$. As given in Equation (6.23), the solution set obtained after each optimization practice is assigned a total voting score of one, meaning that the more solutions there are in one solution set, the less voting power per solution. The rationale behind such design is that we want to grant larger voting power to the solutions in smaller solution sets which are considered to be less

affected by error. Thereafter, the scores assigned are added altogether for each possible damage scenario and the ones with higher voting scores are more likely to give accurate implications about the true structural damage. As a result, we look for indications made by the calibrations rather than a decisive result, which is prone to error and not easy to obtain owing the under-determined nature of fault identification systems. Notably, by keeping one less digit after the decimal point in terms of damage severity α_L, we are able to investigate the voting scores for possible severity ranges, which could further avert practitioners from investigating inaccurate results.

$$\mathrm{vs}(\hat{\boldsymbol{\alpha}}_a) = \mathrm{vs}(\alpha_{aL}, \mathrm{round}(\alpha_{aS})) = \sum_{i=1}^{M} \left| \hat{\boldsymbol{\alpha}}_a \bigcap \mathbf{A}_i \right| / \left| \mathbf{A}_i \right| \tag{6.24}$$

where the rounding operator (i.e., 'round' in Equation (6.24)) approximates a fractional decimal number by one with one less digit, and $\left| \hat{\boldsymbol{\alpha}}_a \bigcap \mathbf{A}_i \right|$ here gives the number of solutions belong to both $\hat{\boldsymbol{\alpha}}_a$ and $\mathbf{A}_i$.

Recall that voting score calculation is designed to endow those solutions more voting power when the solution set is small. We go one step further by withdrawing the voting scores from the solution sets that exceed the average size of all solution sets instead of equally assigning each solution set a voting score of one.

$$\mathrm{vs}_{partial}(\boldsymbol{\alpha}_a) = \sum_{i=1}^{M} \frac{\mathrm{I}(\left| \mathbf{A}_i \right| \le \left| \bar{\mathbf{A}} \right|) \bullet (\left| \boldsymbol{\alpha}_a \bigcap \mathbf{A}_i \right|)}{\left| \mathbf{A}_i \right|} \tag{6.25}$$

$$\mathrm{vs}_{partial}(\hat{\boldsymbol{\alpha}}_a) = \sum_{i=1}^{M} \frac{\mathrm{I}(\left| \mathbf{A}_i \right| \le \left| \bar{\mathbf{A}} \right|) \bullet (\left| \hat{\boldsymbol{\alpha}}_a \bigcap \mathbf{A}_i \right|)}{\left| \mathbf{A}_i \right|} \tag{6.26}$$

where $\mathrm{I}(\left| \mathbf{A}_j \right| \le \left| \bar{\mathbf{A}} \right|)$ is a logic operation that the value of it is 1 if the argument is true and 0 otherwise. By applying either Equation (6.25) or (6.26) for post-processing, a higher level of separation between insightful solutions and trivial solutions could be gained.

In this study, we propose four voting score calculation heuristics (Equations (6.23-6.26)) that essentially serve as four decision making strategies attempting to identify or isolate possible fault scenarios for further inspection. The randomness introduced when selecting response surfaces (Figure 6.3)

as objective functions has desirable characteristics. It not only makes the evaluation scheme robust to outliers, but also gives useful internal estimates of noise such that we can withdraw voting scores from certain solution sets (Equations (6.25)(6.26)). Moreover, it is compatible with parallel computation. Combined with Gaussian process regression and many-objective optimization, the proposed data-assisted many-objective evaluation framework is illustrated in details through numerical and experimental case studies in Section 6.3 and Section 6.4, respectively.

6.3 Numerical case studies

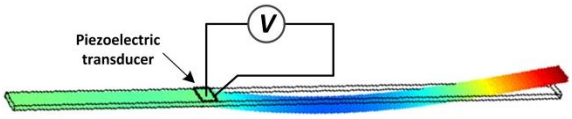
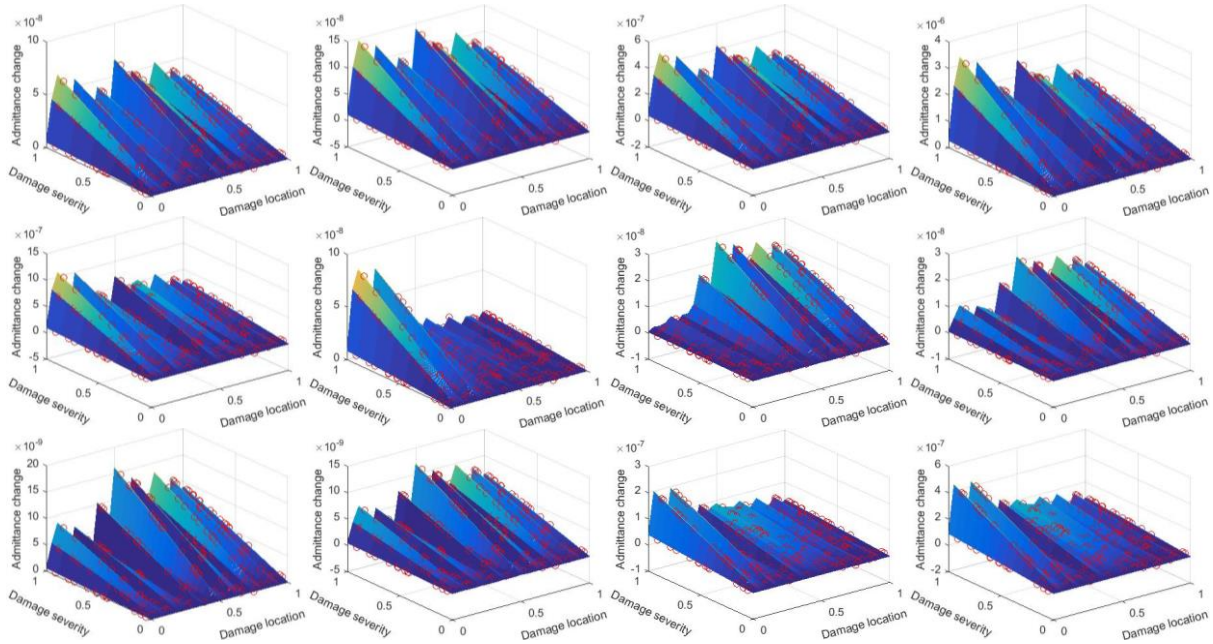


Figure 6.4 Illustration of structure for numerical case studies

In this section, we carry out two numerical case studies using the proposed methodology to gain insights. The structure of interest is an aluminum cantilevered plate (Figure 6.4) with the following properties: length $0.561\,\text{m}$, width $0.01905\,\text{m}$, thickness $0.0014\,\text{m}$, density $2700\,\text{kg/m}^3$, and Young's modulus $68.9\,\text{GPa}$. A piezoelectric transducer is attached to the middle left of the plate, i.e., 0.18 m from the fixed end. The properties of the piezoelectric transducer are: length 0.015 m, width $0.01905\,\text{m}$, thickness $0.0014\,\text{m}$, Young's moduli $Y_{11} = 86\,\text{GPa}$ and $Y_{33} = 73\,\text{GPa}$, density $9500\,\text{kg/m}^3$, piezoelectric constant $-1.0288\times10^9\,\text{V/m}$, and dielectric constant $\beta_{33} = 1.3832\times10^8\,\text{m/F}$. The finite element model of the plate contains 11,250 20-node hexahedron elements, the size of which is smaller than the shortest wavelength of the response involved in this study. The plate is further evenly divided into 25 segments along the lengthwise direction; each is a possible damage location. In structural health monitoring using impedance or admittance measurements, the response changes due to damage occurrence are most evident around the resonant peaks. In the following numerical case studies, we acquire admittance measurements at 40 excitation frequencies around the plate's 14^{th}, 16^{th}, 21^{st} and 23^{rd} natural frequencies. Specifically, the admittance values at 40 evenly distributed excitation frequencies in the ranges 1886.6 Hz to 1890.4 Hz, 2423.7 Hz to 2428.5 Hz, 3694.6 Hz to 3702.0 Hz and 4438.7 Hz to 4447.6 Hz are employed.

Identical for each frequency, 270 randomly generated fault scenarios are emulated for the calibration of impedance response surface. The sampling range is specified as 1 to 25 for location and 0 to 0.1 for severity. A set of impedance measurements is produced by sweeping through the pre-specified excitation frequency points numerically. It is worth noting that in actual implementation, we can directly utilize experimentally acquired measurements in lieu of the numerical ones. The data sampled from the numerical model is contaminated by ± 0.15% standard Gaussian uncertainties to demonstrate the effectiveness of the proposed approach. Here we assume two fault scenarios, i.e., damage occurring at the 13th segment with severity $\alpha_{13} = 0.0600$ (6.00% stiffness loss in 13th segment), and at 22nd segment with severity $\alpha_{22} = 0.0857$ (8.57% stiffness loss), which are randomly selected.

Figure 6.5 plots all 40 impedance response surfaces reconstructed through Gaussian process regression outlined in Section 6.2.2, which serve as 40 objective functions. The two horizontal axes indicate damage location and severity (normalized), and the vertical axis indicates the admittance change measured by the piezoelectric transducer circuit.



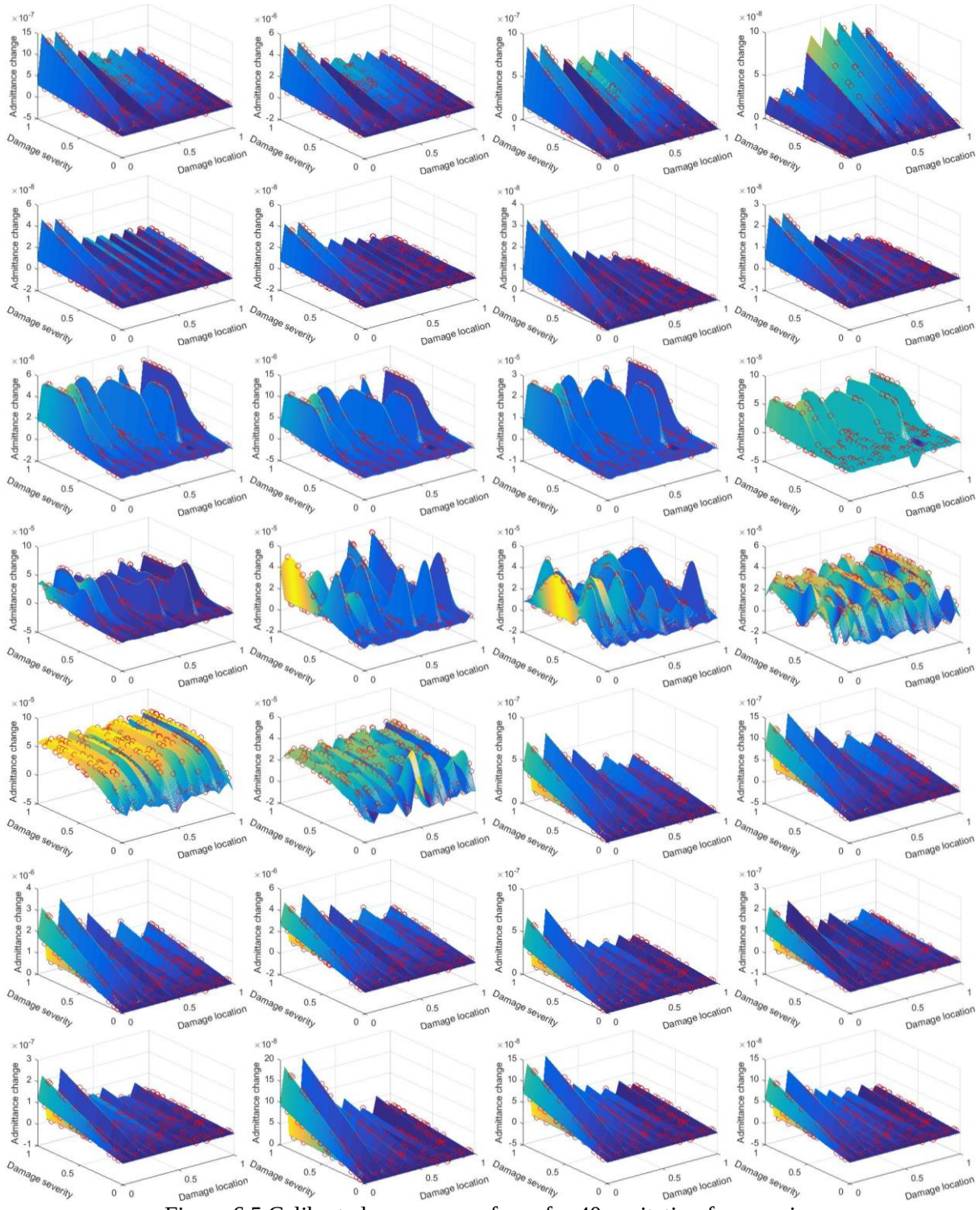


Figure 6.5 Calibrated response surfaces for 40 excitation frequencies (O denotes training point).

Based on the many-objective evaluation approach outlined in Section 6.2.3 (Figure 6.3), for each optimization practice, 10 (N=10) surfaces out of 40 are randomly selected as objective functions, and the

many-objective optimization is executed 30 (M=30) times. The parameter N is selected in accordance with the capacity of the many-objective optimization algorithm implemented, and the value of M should be set as large as possible for robustness. In this study, we use M=30 for illustration. In other words, a total of 30 voting scores are assigned to possible solutions obtained in 30 many-objective optimization practices.

6.3.1 6.00% stiffness loss in 13th segment

We first investigate the case where the 13th segment suffers from 6.00% stiffness loss. Here, the post-processing of the MaOO evaluation results introduced in Section 6.2.3 warrants detailed discussion. After performing the many-objective optimization, we obtain 369 solutions corresponding to 369 possible damage scenarios. Voting score calculation (Equation (6.23)) is then carried out successively accrediting score to each solution as quality quantification. As shown in Figure 6.6, if we consider the solution with the highest voting score as damage identified, it agrees with the actual damage with only 0.0001 discrepancy in stiffness loss ratio.

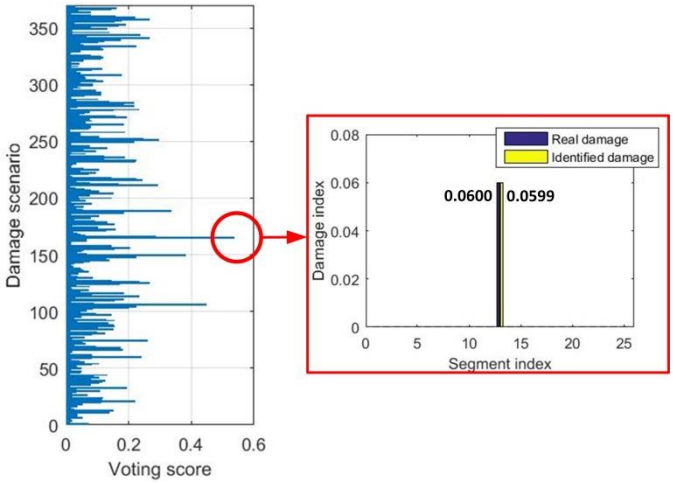


Figure 6.6 Voting scores for 369 fault scenarios and the one with the highest score.

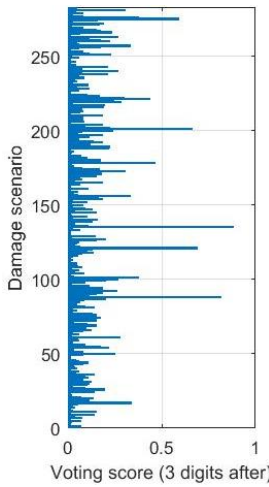


Figure 6.7 Voting scores for 282 damage severity ranges

In the original voting score calculation, four digits are kept in ratio form in terms of damage severity. Then by keeping one less digit after the decimal point following Equation (6.24), we are able to investigate the voting scores for possible severity ranges, as illustrated in Figure 6.7. Out of 282 damage severity ranges, some are more distinct from the others compare to the results shown in Figure 6.6, which could be considered as candidate solutions. Figure 6.8 visualizes the severity range with the highest voting score, in which the induced damage is indeed included.

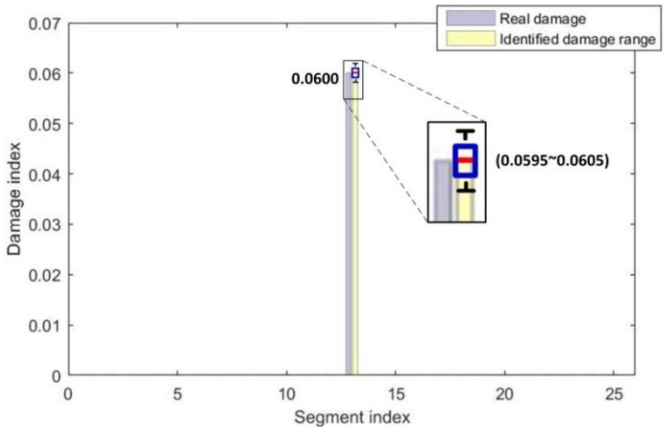
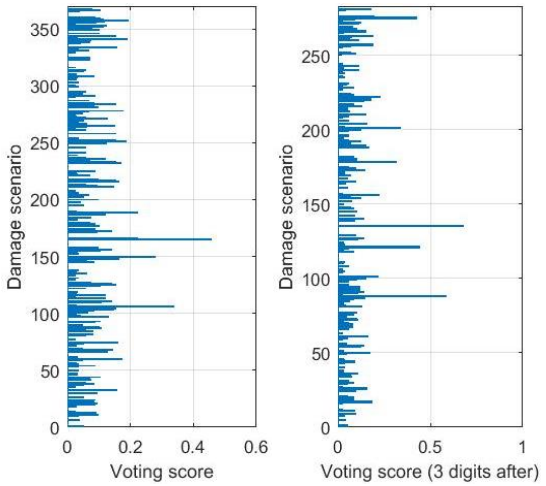


Figure 6.8 Identified damage range (the severity range with the highest voting score)

We then withdraw the voting scores from the solution sets that exceed the average size of all solution sets as suggested in Equation (6.25). A total of 16 voting scores are to be distributed among possible solutions found by the many-objective optimization algorithm. As shown in Figure 6.9(a), some of the

fault scenarios are now affiliated with zero voting scores. Comparably, the solutions with higher voting scores are more significantly separated. Similarly, the voting scores for severity ranges can be inspected by grouping certain damage severities together (Equation (6.26)) and, as illustrated in Figure 6.9(b), a greater separation is achieved due to the aggregation of voting scores among similar solutions. The purpose of getting higher level of separation hinges on the fact that such fault identification scheme is under-determined due to deficient measurements, uncertainties and errors. Thus, multiple solutions are expected. The proposed data-assisted many-objective evaluation endeavors to isolate a small number of possible solutions by their voting scores for further inspections.



(a) (b)

Figure 6.9 (a) Partial voting scores for 369 damage scenarios (b) Partial voting scores for 282 damage severity ranges

Table 6.2 lists the fault scenarios with top five highest voting scores calculated following four different heuristics introduced in Section 6.2.3. As shown in the table, the ones with the highest scores all match or cover the true fault scenario. The percentage of voting score out of all voting scores being allotted is also reported in Table 6.2. For the result that indicates true damage the best, the voting score percentage increases either by grouping severities to severity ranges or assign zero voting scores to large solution sets, meaning a higher level of separation or a higher level of confidence is achieved.

Nevertheless, when prior knowledge is unavailable, solutions with relatively higher scores all should be considered as possible fault scenarios. The proposed voting score scheme filters out most scenarios.

Therefore, even though one single certain solution is hard, if not impossible, to obtain, only a few need to be examined with the help of many-objective evaluation, and the one with the highest voting score is more likely to match the true fault scenario.

Table 6.2 Top five fault scenarios with highest voting scores

I: Voting score (Equation (6.23))			
Fault scenario		Voting score	Score % 30 overall
Segment	Severity		
13	0.0599	0.5351	1.784%
10	0.0522	0.4474	1.491%
12	0.0532	0.3822	1.274%
15	0.0505	0.3340	1.113%
17	0.0880	0.2922	0.974%
II: Voting score for severity range (Equation (6.24))			
Fault scenario		Voting score	Score % 30 overall
Segment	Severity range		
13	0.0595~0.0605	0.8822	2.941%
10	0.0515~0.0525	0.8137	2.712%
12	0.0525~0.0535	0.6881	2.294%
17	0.0875~0.0885	0.6634	2.211%
25	0.0575~0.0585	0.5896	1.965%
III: Partial voting score (Equation (6.25))			
Fault scenario		Voting score	Score % 16 overall
Segment	Severity		
13	0.0599	0.4579	2.862%
10	0.0522	0.3373	2.108%
12	0.0532	0.2790	1.744%
15	0.0505	0.2239	1.399%
13	0.0602	0.2215	1.384%
IV: Partial voting score for severity range (Equation (6.26))			
Fault scenario		Voting score	Score % 16 overall
Segment	Severity range		
13	0.0595~0.0605	0.6794	4.246%
10	0.0515~0.0525	0.5866	3.666%
12	0.0525~0.0535	0.4434	2.771%
25	0.0575~0.0585	0.4251	2.657%
17	0.0875~0.0885	0.3400	2.125%

6.3.2 8.57% stiffness loss in 22nd segment

To further demonstrate the proposed approach, we investigate a second numerical case where the 22nd segment suffers from 8.57% stiffness loss. After performing the many-objective optimization, we first

obtain 491 possible fault scenarios (Figure 6.10(a)), which can be grouped into 365 severity ranges following Equation (6.24) (Figure 6.10(b)). Aiming at separating the solutions with relatively higher voting scores, the post-processing heuristics given in Equations (6.25) and (6.26) are implemented here as shown in Figure 6.10(c) and 6.10(d), respectively. The results with top five highest voting scores are ranked in Table 6.3. Similar to the results reported in Section 6.3.1, in this case study, the ones with the highest scores all agree with or cover the true fault scenario.

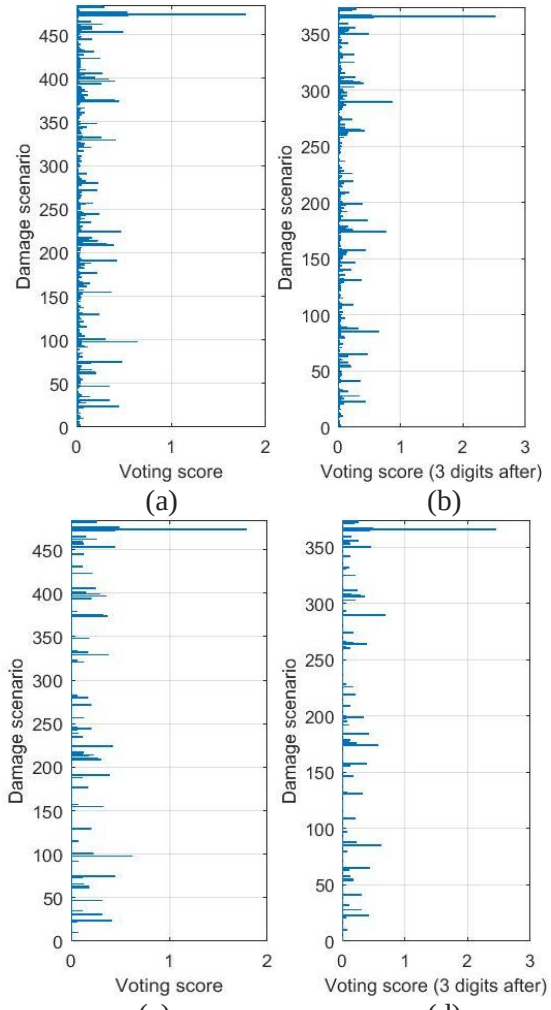


Figure 6.10 (a) Voting scores for 491 fault scenarios (b) Voting scores for 365 damage severity ranges (c) Partial voting scores for 491 fault scenarios (d) Partial voting scores for 365 damage severity ranges

Table 6.3 Top five fault scenarios with highest voting scores

I: Voting score (Equation (6.23))		
Fault scenario	Voting	Score %

Segment	Severity	score	30 overall
22	0.0856	1.0044	3.348%
25	0.0813	0.7574	2.525%
25	0.0923	0.6529	2.176%
22	0.0843	0.4427	1.476%
13	0.0946	0.4389	1.463%
II: Voting score for severity range (Equation (6.24))			
Fault scenario		Voting score	Score % 30 overall
Segment	Severity range		
22	0.0855~0.0865	1.4931	4.977%
25	0.0915~0.0925	0.8741	2.914%
25	0.0805~0.0815	0.7573	2.524%
15	0.0675~0.0685	0.7321	2.440%
13	0.0875~0.0885	0.6866	2.289%
III: Partial voting score (Equation (6.25))			
Fault scenario		Voting score	Score % 17 overall
Segment	Severity		
22	0.0856	1.0044	5.908%
25	0.0813	0.7433	4.372%
25	0.0923	0.5045	2.968%
22	0.0843	0.4027	2.369%
22	0.0863	0.3533	2.078%
IV: Partial voting score for severity range (Equation (6.26))			
Fault scenario		Voting score	Score % 17 overall
Segment	Severity range		
22	0.0855~0.0865	1.444	8.494%
25	0.0805~0.0815	0.7433	4.372%
25	0.0915~0.0925	0.7257	4.269%
22	0.0835~0.0845	0.5675	3.338%
13	0.0935~0.0945	0.5674	3.338%

Here we also investigate the effectiveness of the proposed voting score calculation (Equation (6.23)) in discriminating possible damage scenarios. The idea of voting has also been used in ensemble learning such as random forest (Breiman, 2001) and pattern recognition (Lam and Suen, 1997) to combine different sets of result where majority voting is implemented. Compared to the proposed voting score strategy, majority voting considers the one damage scenario that appears the most in all solution sets as the indication of true damage. Figure 6.11 compares the solution with the highest voting score, which concur with the true damage scenario, to the solution that appears the most in all solution sets. As revealed in Table 6.4, the voting score calculation successfully re-adjusts the weighting among all solutions. The voting score heuristics manage to rank them essentially based on their quality and thus

have better performance in terms of identifying the true damage. Indeed, the solution with the highest voting score is not only among the solutions that appear the most in the results of many-objective optimizations, but also is less affected by error trade-offs because it appears mostly in small solution sets which are considered more insightful with less conflicting objective functions. After all, the objective functions should not be contradicted with each other ideally without error.

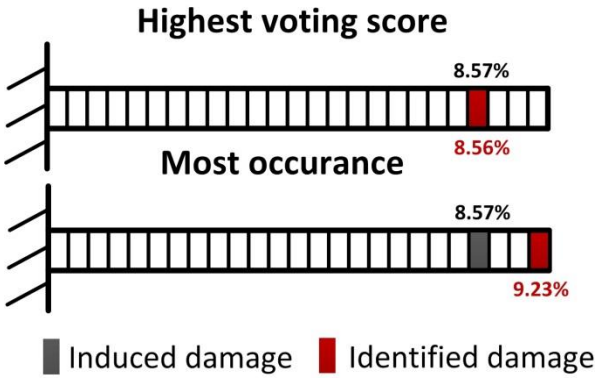


Figure 6.11 The fault scenario with the highest voting score vs. the damage scenario with the most occurrence

Table 6.4 Top five fault scenarios: highest voting scores vs. most occurrences

I: Proposed voting score strategy		
Damage scenario		Voting score
Segment	Severity	30 overall
22	0.0856	1.0044
25	0.0813	0.7574
25	0.0923	0.6529
22	0.0843	0.4427
13	0.0946	0.4389
II: Majority voting (most occurrence)		
Damage scenario		Occurrence
Segment	Severity	1289 overall
25	0.0923	17
15	0.0682	16
13	0.0946	15
22	0.0856	14
13	0.0880	14

6.4 Experimental validation

In this section, experimental case studies using physical measurements of piezoelectric admittance are

carried out. The experimental setup, geometry measures and material parameters are consistent with those used in the numerical analysis in Section 6.3. Figure 6.12 shows the experimental setup. To obtain the admittance of the piezoelectric circuit, a resistor of $100\,\Omega$ is serially-connected to the transducer to measure the voltage drop, which is further used to extract the current in the circuit. An Agilent 35670A signal analyzer is employed, where the source channel is used to generate sinusoidal voltage sent to piezoelectric transducer denoted as V_{in}, and the output voltage across the resistor is recorded as V_{out}. Hence, the admittance can be obtained as $Y = I/V_{\text{in}} = V_{\text{out}}/R_s V_{\text{in}}$. In experimental case studies, we acquire measurement samples using 18 excitation frequencies around the plate's 14th and 21st natural frequencies. That is, 10 evenly distributed frequencies from the range 1886.6 Hz to 1890.4 Hz and 8 evenly distributed frequencies from the range 3696.2 Hz to 3702.0 Hz are acquired.

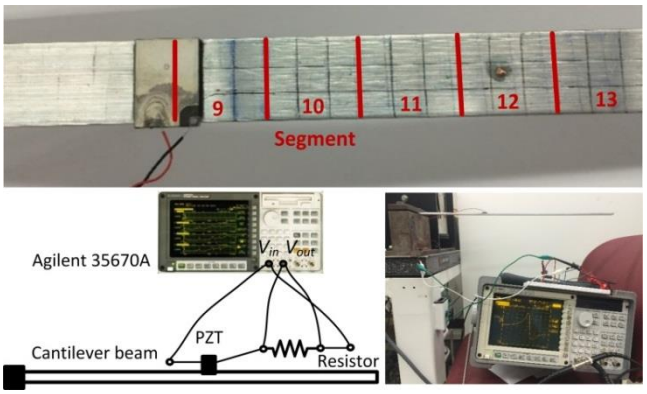


Figure 6.12 Experiment Setup

Identical for each frequency, 150 randomly generated damage scenarios are emulated for impedance response surface calibration using the corresponding numerical model. Figure 6.13 illustrates all 18 impedance response surfaces reconstructed by Gaussian process regression. In order to reduce the unwanted variations and uncertainties in this case illustration, instead of disassembling and cutting the plate to reduce the local stiffness, we add small masses to emulate the damage occurrence. Mathematically, adding a small mass can result in the same resonant frequency shift and admittance change as a local stiffness reduction would. In the first experiment, a 0.6 g mass is attached to the 14th segment of the plate, which causes admittance change equivalent to a 0.28% local stiffness loss. In the

second experiment, the same mass is attached to the 12th segment, which is equivalent to a 0.16% local stiffness loss.

Based on the methodology proposed, the many-objective optimization is executed 10 times, and for each optimization execution, 10 surfaces out of 18 are randomly selected as objective functions. In other words, a total of 10 voting scores are assigned to solutions obtained.

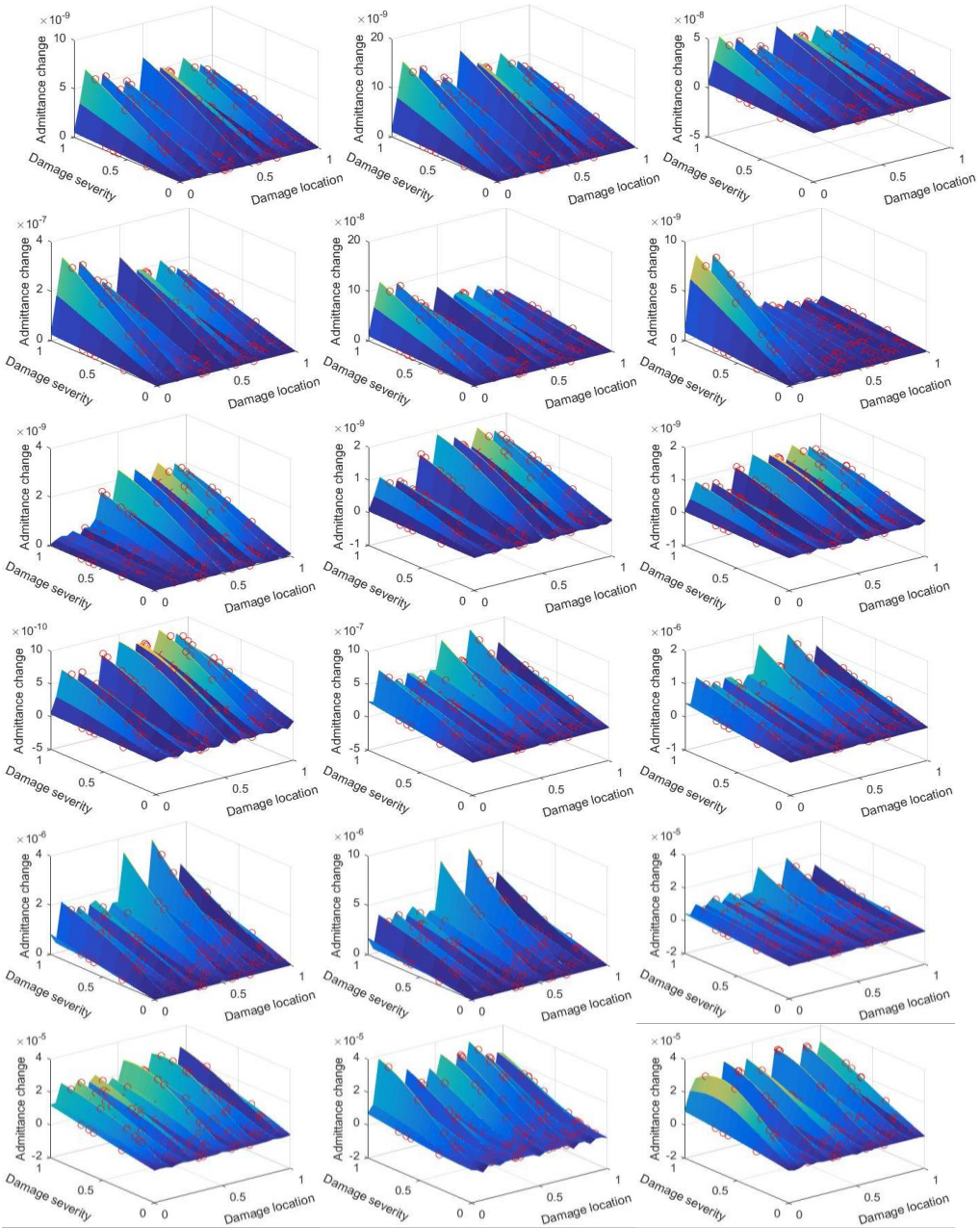
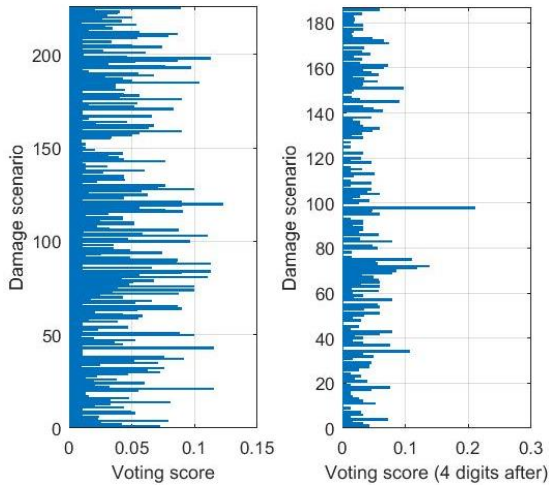


Figure 6.13 Calibrated response surfaces for 18 excitation frequencies from small to large (O denotes training point)

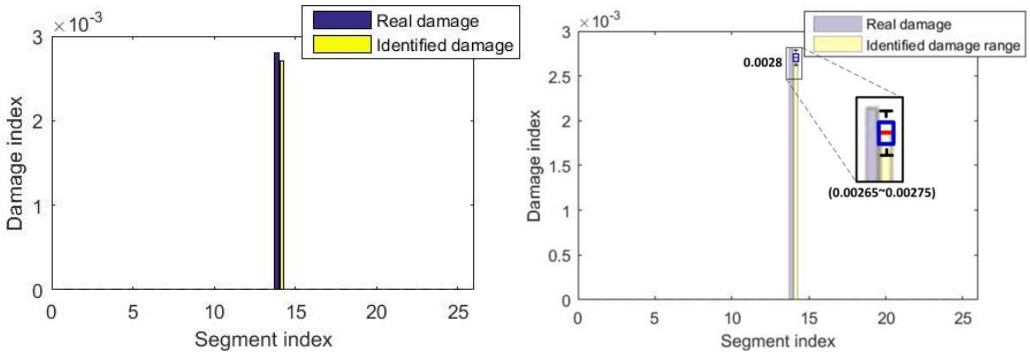
6.4.1 0.28% stiffness loss in 14th segment

We first perform the experimental case study where the 14th segment is subjected to a 0.28% equivalent stiffness loss. Figure 6.14(a) plots the voting scores (Equation (6.23)) for all possible fault scenarios after many-objective optimization. As uncertainties such as modeling error and measurement error are present inevitably in experimental case studies, we cannot easily distinguish the better solutions. However, by examining the solutions based on the severity ranges they fall into following the heuristic given in Equation (6.24) (Figure 6.14(b)), a few solutions stand out. As shown in Figure 6.15, if we consider the solutions with the highest voting scores as damage identified or the damage severity range identified, accurate predictions can be achieved. In practice, the solutions with relatively higher voting scores should be considered as candidates. Such candidate set provided by the proposed evaluation approach serves as the starting point for further inspections, which streamlines the typical procedure of inspection and maintenance in engineering practice.



(a) (b)

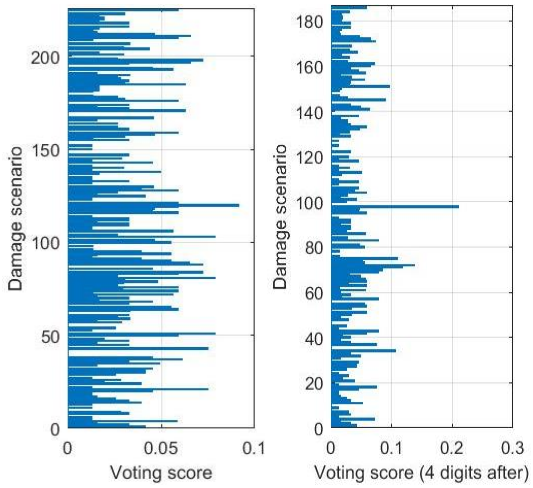
Figure 6.14 (a) Voting scores for 225 fault scenarios (b) Voting scores for 186 damage severity ranges



(a) (b)

Figure 6.15 (a) Identified damage (the damage scenario with the highest voting score) (b) Identified damage range (the severity range with the highest voting score)

Next, by assigning zero voting scores to the solution sets that exceed the average size of all solution sets (Equation (6.25)), we can probe some of the solutions provided by certain solution sets which are considered of better quality. As shown in Figure 6.16, a higher level of distinction is achieved among solutions. The results are reported in Table 6.5 where the top five damage scenarios with highest voting scores are demonstrated.



(a) (b)

Figure 6.16 (a) Partial voting scores for 225 fault scenarios (b) Partial voting scores for 186 damage severity ranges.

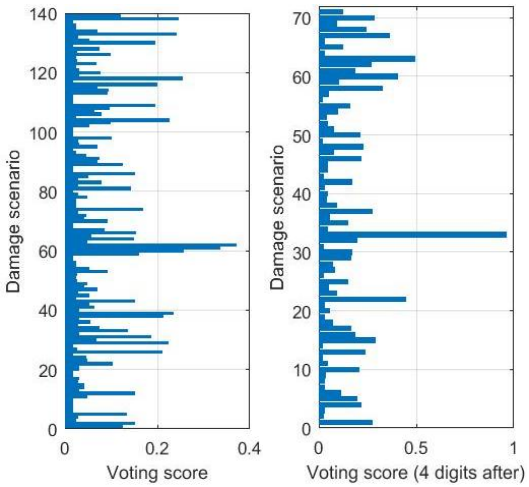
Table 6.5 Top five fault scenarios with highest voting scores

I: Voting score (Equation (6.23))			
Fault scenario		Voting score	Score % 10 overall
Segment	Severity		
14	0.00271	0.1223	1.223%
5	0.00609	0.1153	1.153%

3	0.00294	0.1153	1.153%
22	0.00492	0.1126	1.126%
11	0.00719	0.1126	1.126%
II: Voting score for severity range (Equation (6.24))			
Fault scenario		Voting score	Score % 10 overall
Segment	Severity range		
14	0.00265~0.00275	0.3016	3.016%
11	0.00715~0.00725	0.2078	2.078%
11	0.00705~0.00715	0.1882	1.882%
21	0.00915~0.00925	0.1777	1.777%
5	0.00515~0.00525	0.1671	1.671%
III: Partial voting score (Equation (6.25))			
Fault scenario		Voting score	Score % 6 overall
Segment	Severity		
14	0.00271	0.0917	1.528%
6	0.00458	0.0789	1.315%
12	0.00645	0.0787	1.312%
11	0.00622	0.0787	1.312%
5	0.00609	0.0747	1.245%
IV: Partial voting score for severity range (Equation (6.26))			
Fault scenario		Voting score	Score % 6 overall
Segment	Severity range		
14	0.00265~0.00275	0.2099	3.498%
11	0.00715~0.00725	0.1372	2.287%
11	0.00705~0.00715	0.1176	1.960%
11	0.00795~0.00805	0.1102	1.837%
5	0.00515~0.00525	0.1064	1.773%

6.4.2 0.16% Stiffness Loss in 12th Segment

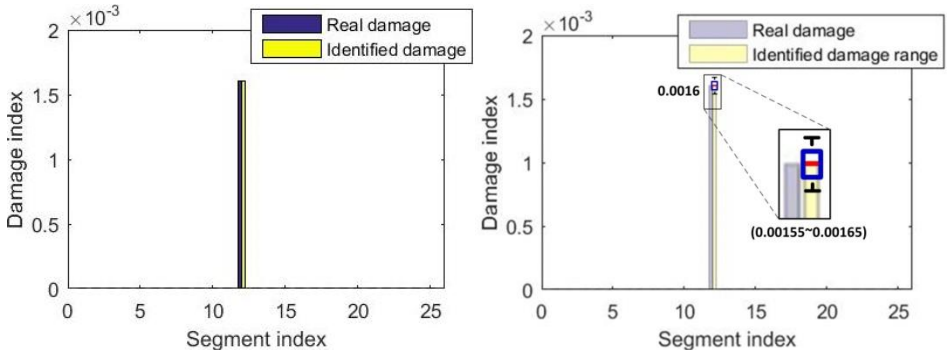
The second experimental study concerns the case where the 12th segment is subjected to a 0.16% equivalent stiffness loss. Figure 6.17 plots the voting scores for all 139 possible fault scenarios and 71 severity ranges after the many-objective evaluation following Equation (6.23) and (6.24). It is noticed that a small set of solutions clearly maintains a margin over the rest of the solutions in terms of voting score. As shown in Figure 6.18, the solution with the highest voting score delivers close indication about the health condition of the structure. And if we consider the severity range with the highest voting score as the identified damage severity range, it also covers the true damage scenario.



(a) (b)

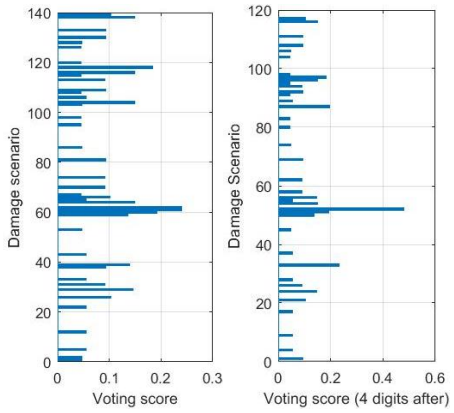
Figure 6.17 (a) Voting scores for 139 fault scenarios (b) Voting scores for 71 damage severity ranges

Similar to the preceding case studies, zero voting scores are assigned to the solution sets that exceed the average size (Equation (6.25)), which produces a more polarized voting score distribution shown in Figure 6.19. As can be seen in Table 6.6, for all four post-processing means, the ones with the highest voting scores all make accurate implications of the health condition of the structure.



(a) (b)

Figure 6.18 (a) Identified fault (the fault scenario with the highest voting score) (b) Identified damage range (the severity range with the highest voting score)



(a) (b)

Figure 6.19 (a) Partial voting scores for 139 fault scenarios (b) Partial voting scores for 71 damage severity ranges

Table 6.6 Top five fault scenarios with highest voting scores

I: Voting score (Equation (6.23))			
Damage scenario		Voting score	Score % 10 overall
Segment	Severity		
12	0.00161	0.3696	3.696%
12	0.00157	0.3355	3.355%
12	0.00154	0.2554	2.554%
22	0.00161	0.2536	2.536%
25	0.00027	0.2446	2.446%
II: Voting score for severity range (Equation (6.24))			
Damage scenario		Voting score	Score % 10 overall
Segment	Severity range		
12	0.00155~0.00165	0.7050	7.050%
8	0.00135~0.00145	0.4467	4.467%
21	0.00145~0.00155	0.3232	3.232%
1	0.00105~0.00115	0.2734	2.734%
22	0.00155~0.00165	0.2705	2.705%
III: Partial voting score (Equation (6.25))			
Damage scenario		Voting score	Score % 4 overall
Segment	Severity		
12	0.00160	0.2395	5.988%
12	0.00157	0.2395	5.988%
12	0.00154	0.1919	4.798%
22	0.00161	0.1840	4.600%
25	0.00027	0.1486	3.715%
IV: Partial voting score for severity range (Equation (6.26))			
Damage scenario		Voting score	Score % 4 overall
Segment	Severity range		
12	0.00155~0.00165	0.4791	11.978%
8	0.00135~0.00145	0.2316	5.790%
21	0.00145~0.00155	0.1962	4.910%
12	0.00145~0.00155	0.1919	4.798%
22	0.00155~0.00165	0.1839	4.598%

Finally, the four different means of handling voting scores used in case studies are compared in Figure 6.20. The voting power of the best solution in each case study is quantified in the form of voting score percentage, which is the percentage of its voting score out of available voting scores being distributed. As illustrated, by either grouping the damage severities to severity intervals or assigning voting scores to only certain solution sets, the confidence of the implications made by the proposed approach, which is directly related to the voting power, is increased. In practice, as we want to inspect only a small number of damage scenarios in maintenance, the overall approach proposed in this study can help to isolate a small set of the solutions that are more related to the health condition of the structure through its data-assisted analysis. Instead of seeking for one deterministic solution that could be misguiding, the approach proposed in this research utilizes training data to analyze and identify probable fault scenarios that serve as guidance for further examination through heterogeneous sensing and inspection.

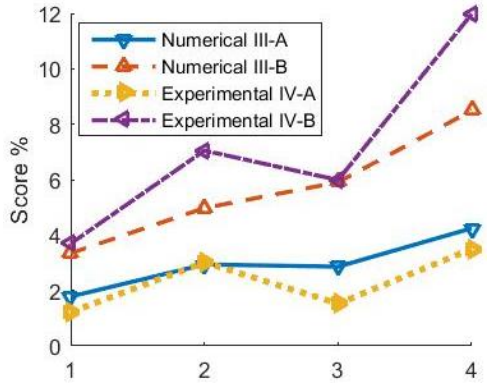


Figure 6.20 Voting score percentage of the best solution in each case study as four post-processing techniques adopted (Ticks on the horizontal axis represent: voting scores for damage scenarios, voting scores for damage severity ranges, partial voting scores for damage scenarios and partial voting scores for damage severity ranges, respectively from left to right)

6.5 Concluding remarks

This research presents a data-assisted approach for structural fault identification through Gaussian process-based impedance response calibration and many-objective evaluation. To address the fundamental challenges posed by the under-determined problem formulation and model-based sensitivity

approximation, we cast the damage identification problem into a many-objective optimization by reconstructing impedance response surfaces as objective functions utilizing training data. The optimization problem is then tackled by an ε-dominance enabled many-objective simulated annealing algorithm. As many solutions are expected in many-optimization practices, a voting score calculation procedure is developed and applied after to quantify and identify the solutions that could make better implication about the health condition of the structure. The numerical case studies and experimental case studies demonstrate that the proposed approach is capable of obtaining a small set of solutions based on their voting scores that could provide accurate implication about the health condition of the interested structure. The proposed scheme is inherently malleable and can be applied to either model-based or model-free fault identification systems wherever data is available. The combination of Gaussian process-based calibration, many-objective optimization, and voting score calculation can be extended to a variety of inverse analysis problems.

Chapter 7. Preprocessing-Free Gear Fault Diagnosis Using Small Datasets with Deep Convolutional Neural Network-Based Transfer Learning

Early diagnosis of gear transmission has been a significant challenge, because gear faults occur primarily at microstructure or even material level but their effects can only be observed indirectly at a system level. The performance of a gear fault diagnosis system depends significantly on the features extracted and the classifier subsequently applied. Traditionally, fault-related features are extracted and identified based on domain expertise through data preprocessing which are system-specific and may not be easily generalized. On the other hand, although recently the deep neural networks based approaches featuring adaptive feature extractions and inherent classifications have attracted attention, they usually require a substantial set of training data. Aiming at tackling these issues, this paper presents a deep convolutional neural network-based transfer learning approach. The proposed transfer learning architecture consists of two parts; the first part is constructed with a pre-trained deep neural network that serves to extract the features automatically from the input, and the second part is a fully connected stage to classify the features that needs to be trained using gear fault experimental data. Case analyses using experimental data from a benchmark gear system indicate that the proposed approach not only entertains preprocessing free adaptive feature extractions, but also requires only a small set of training data.

7.1 Introduction

Condition monitoring and fault diagnosis play essential role in ensuring the safe and sustainable operations of modern machinery systems. Gearbox, as one common component used in those systems, is prone to fault condition or even failure, because of the severe working condition with high mechanical loading and typically long operational time. Currently, vibration signals are most widely used to infer the health condition of gear system, because they contain rich information and can be easily measured using off-the-shelf, low-cost sensors. Indeed, gear vibration signals contain three components: periodic meshing frequencies, their harmonics, and random noise. For a healthy gear system, the meshing frequencies and their harmonics dominate the vibration response. Fault conditions cause additional

dynamic effects.

The practice of fault diagnosis of gear system using vibration signals has proved to be a very challenging subject. The mainstream of gear condition monitoring is built upon various feature extraction methods that are manual and empirical in nature (Kang et al, 2001; Randall, 2011; Marquez et al, 2012). Generally, a certain signal processing technique is applied to vibration signals to identify fault-related features that are selected based on engineering judgment. Subsequently, a classifier is developed and applied to new signals to predict fault occurrence in terms of type and severity. There have been extensive and diverse attempts in manually and empirically identifying and extracting useful features from gear vibration signals, which fall into three main categories: time-domain analysis (Zhou et al, 2008; Parey and Pachori, 2012), frequency domain-analysis (Fakhfakh et al, 2005; Li et al, 2015; Wen et al, 2015) and time-frequency analysis (Tang et al, 2010; Chaari et al, 2012; Yan et al, 2014; Chen and Feng, 2017; Zhang and Tang, 2018). Time-domain statistical approaches can capture the changes in amplitude and phase modulation caused by faults (Parey and Pachori, 2012; Pachaud et al, 1997). In comparison, spectrum analysis may extract the features more easily to detect distributed faults with clear sidebands (Fakhfakh et al, 2005; Wen et al, 2015; Qian and Chen 1999). To deal with noise and at the same time utilize the transient components in vibration signals, many efforts have focused on joint time-frequency domain analysis utilizing Wigner-Ville distribution (Tang et al, 2010; Baydar and Ball, 2001), short time Fourier transform (Chaari et al, 2012; Bartelmus and Zimroz, 2009), and various wavelet transforms (Yan et al, 2014; Lin and Zuo 2003). The time-frequency distribution in such analysis can in theory lead to rich analysis results regarding the time- and frequency-related events in signals.

Although the manual and empirical methods of feature extraction have seen various levels of successes, obviously their effectiveness is hinged upon the specific features adopted in the diagnostic analysis. It is worth emphasizing that the choices of features as well as the often-applied signal preprocessing techniques are generally based on domain expertise and subjective decisions on a specific gear system. For example, while wavelet transforms have been popular and it is well known that each wavelet coefficient can be interpreted as the energy concentration at a specific time-frequency point, it is

evident from large amount of literature that there does not seem to be a consensus on what kind of wavelet to use for gear fault diagnosis. This should not come as a surprise. On one hand gear faults occur primarily at microstructure or even material level but their effects can only be observed indirectly at a system level; consequently there exists a many-to-many relationship between actual faults and the observable quantifies (i.e., features) for a given gear system (Lu et al, 2012). On the other hand, different gear systems have different designs which lead to very different dynamic characteristics. As such, the result on features manually selected and, to a large extent, the methodology employed to extract these features for one gear system design may not be easily extrapolated to a different gear system design.

Fundamentally, condition monitoring and fault diagnosis of gear systems belongs to the general field of pattern recognition. The advancements in related algorithms along with the rapid enhancement of computational power have trigged the wide spread of machine learning techniques to various applications. Most recently, deep neural network-based methods are progressively being investigated. When the parameters of a deep neural network are properly trained by available data, representative features can be extracted in a hierarchy of conceptual abstractions, which are free of human interference compared to manual selection of features. Some recent studies have adopted such type of approaches in gear fault diagnosis, aiming at identifying features implicitly and adaptively and then classifying damage/fault in an automated manner with minimal tuning. For example, Zhang et al (2015) developed a deep learning network for degradation pattern classification and demonstrated the efficacy using turbofan engine dataset. Li et al (2016) proposed a deep random forest fusion technique for gearbox fault diagnosis which achieves 97.68% classification accuracy. Weimer et al (2016) examined the usage of deep convolutional neural network for industrial inspection and demonstrated excellent defect detection results. Ince et al (2016) developed a fast motor condition monitoring system using a 1-D convolutional neural network with a classification accuracy of 97.4%. Abdeljaber et al (2017) performed real-time damage detection using convolutional neural network and showcased satisfactory efficiency.

In this research, aiming at advancing the state-of-the-art, we present a deep neural network-based transfer learning approach utilizing limited time-domain data for gearbox fault diagnosis. One-

dimensional time-domain data of vibration responses related to gear fault patterns are converted into graphical images as input. The approach inherits the non-biased nature of neural networks that can avoid the manual selection of features. Meanwhile, the issue of limited data is overcome by formulating a new neural network architecture that consists of two parts. Massive image data (1.2 million) from ImageNet (http://www.image-net.org/challenges/LSVRC/2010/) are used first to train an original deep neural network model, denoted as neural network A. The parameters of neural network A are transferred (copied) to the new architecture as the first part. The second part of the architecture, an untrained neural network B, accommodates the gear fault diagnosis task and is further trained using experimentally generated gear fault data. Unlike traditional neural networks, the training set of transfer learning do not necessarily subordinate to the same category or from the same physical background (Yang et al, 2018). As to be demonstrated later, with this new architecture, highly accurate gear fault diagnosis can be achieved using limited time-domain data directly without involving any subjective preprocessing techniques to assist feature extraction. The rest of this chapter is organized as follows. In Section 7.2, building upon convolutional neural network and transfer learning, we develop the specific architecture for gear fault diagnosis. In Section 7.3, experimental data are analyzed using the proposed approach with uncertainties and noise; comparisons with respect to different approaches are conducted as well. Concluding remarks are summarized in Section 7.4.

7.2 Transfer learning for gear fault diagnosis

The proposed transfer learning approach is built upon deep convolutional neural network. Deep neural networks have enjoyed great success but require a substantial amount of training instances for satisfactory performance. In this section, for the sake of completeness in presentation we start from the essential formulations of convolutional neural network and transfer learning, followed by the specific architecture developed for gear fault diagnosis with limited training data.

7.2.1 Convolutional neural networks (CNNs)

Convolutional Neural Networks (CNNs) are a class of biologically inspired neural networks featuring

one or multiple convolutional layers that simulate human visual system (LeCun et al, 1990). In recent years, due to the enhancement in computational power and the dramatic increase in the amount of data available in various applications, CNNs-based methods have shown significant improvements in performance and thus have become the most popular class of approaches for pattern recognition tasks such as image classification (Krizhevsky et al, 2012), natural language processing (Kim, 2014), recommending systems (Van den Oord et al, 2013) and fault detection (Ince et al, 2016). CNNs learn how to extract and recognize characteristics of the target task by combining and stacking convolutional layers, pooling layers and fully connected layers in its architecture. Figure 7.1 illustrates a simple CNN with an input layer to accept input images, a convolutional layer to extract features, a ReLU layer to augment features through non-linear transformation, a max pooling layer to reduce data size, and a fully connected layer combined with a softmax layer to classify the input to pre-defined labels. The parameters are trained through a training dataset and updated using back propagation algorithm to reflect the features of the task that may not be recognized otherwise. The basic mechanism of layers in CNNs is outlined as follows.

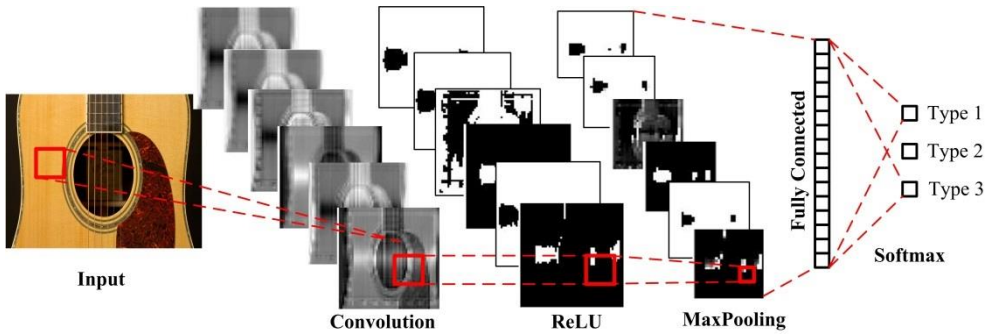


Figure 7.1 An example of convolutional neural network.

Convolutional layer Each feature map in the convolutional layer shown in Figure 1 is generated by a convolution filter. Generally, the input and convolution filters are tensors of size $m \times n$ and $p \times q \times K$ (K is the number of filter used), respectively. Stride (i.e., step size of the filter sliding over input) is set to 1 and padding (i.e., the number of rows and columns to insert around the original input) is set to 0. The

convolution operation can be expressed as,

$$y_{d_1,d_2,k} = \sum_{i=0}^{p}\sum_{j=0}^{q} x_{d_1 i, d_2 j} \times f_{i,j,k} \tag{7.1}$$

where *y*, *x* and *f* denote the element in feature map, input and convolution filter, respectively. $f_{i,j,k}$ represents the element on the *i*-th column and *j*-th row for filter *k*. $y_{d_1,d_2,k}$ is the element on the d_1-th column and d_2-th row of feature map *k*. And $x_{d_1 i, d_2 j}$ refers to the input element on the *i*-th column and *j*-th row of the stride window specified by d_1 and d_2. Equation (7.1) gives a concise representation of the convolution operation when the input is 2-demensional, and stride and padding are 1 and 0. Higher dimension convolution operations can be conducted in a similar manner. To be more evocative, suppose the input image can be represented by a 4×7 matrix and the convolution kernel is a 3×3 identity matrix. As we take kernel and stride it over the image matrix, dot products are taken in each step and recorded in a feature map matrix (Figure 7.2). Such operation is called convolution. In CNNs, multiple convolution filters are used in a convolutional layer, each acquiring a feature piece in its own perspective from the input image specified by the filter parameters. Regardless of what and where a feature appears in the input, the convolutional layer will try to characterize it from various perspectives that have been tuned automatically by the training dataset.

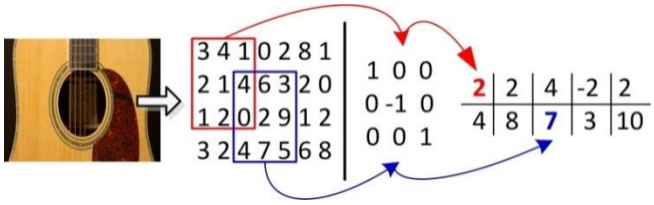


Figure 7.2 Illustration of convolution operation.

ReLU layer In CNNs, ReLU (rectified linear units) layers are commonly used after convolutional layers. In most cases, the relationship between the input and output is not linear. While the convolution operation is linear, the ReLU layer is designed to take non-linear relationship into account, as shown in the equation below,

$$\bar{y} = \max(0, y) \tag{7.2}$$

The ReLU operation is applied to each feature map and returns an activation map (Figure 7.3). The depth of the ReLU layer equals to that of the convolutional layer.

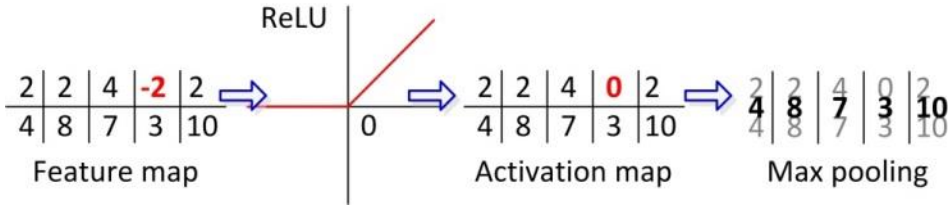


Figure 7.3 Illustration of ReLU and max pooling.

Max pooling layer Max pooling down-samples a sub-region of the activation map to its maximum value,

$$\hat{y} = \max_{L_1 \le i \le U_1,\, L_2 \le i \le U_2} \bar{y}_{i,j} \tag{7.3}$$

where $L_1 \le i \le U_1$ and $L_2 \le j \le U_2$ define the sub-region. The max pooling layer not only makes the network less sensitive to location changes of a feature but also reduces the size of parameters, thus alleviates computational burden and controls overfitting.

7.2.2 Transfer learning

CNNs are powerful tools, and the performance can generally be improved by up-scaling the CNN equipped. The scale of a CNN concurs with the scale of the training dataset. Naturally, the deeper the CNN, the more parameters need to be trained, which requires a substantial amount of valid training samples. Nevertheless, in gear fault diagnosis, the training data is not as sufficient as that of data-rich tasks such as natural image classification. In fact, it is impractical to collect physical data from each failure type and especially severity since the severity level is continuous in nature and there are infinitely many possible fault profiles.

Figure 7.4 illustrates a representative relationship between data size and performance for different learning methods. While the performance of a large-scale CNN has the potential to top other methods, it is also profoundly correlated with the size of training data. Transfer learning, on the other hand, is

capable of achieving prominent performance commensurate with large scale CNNs using only a small set of training date (Shie et al, 2015; Zhang et al, 2017). By applying knowledge and skills (in the form of parameters) learned and accumulated in previous tasks that have sufficient training data, transfer learning provides a possible solution to improve the performance of a neural network when applied to a novel task with small training dataset. Classic transfer learning approaches transfer (copy) the first n layers of a well-trained network to the target network of layer $m > n$. Initially, the last $(m-n)$ layers of the target network are left untrained. They are trained subsequently using the training data from the novel task. Let the training datasets from the previous task $\mathbf{D}_{\text{pre}}$ and the novel task $\mathbf{D}_{\text{nov}}$ be represented as

$$\mathbf{D}_{\text{pre}} = \left\{\mathbf{X}_{\text{pre}}, \mathbf{L}_{\text{pre}}\right\},\ \mathbf{D}_{\text{nov}} = \left\{\mathbf{X}_{\text{nov}}, \mathbf{L}_{\text{nov}}\right\} \tag{7.4a, b}$$

where $\mathbf{X}$ is the input and $\mathbf{L}$ is the output label. The CNNs for both tasks can then be regarded as, $\hat{\mathbf{L}}_{\text{pre}} = \text{CNN}_{\text{pre}}(\mathbf{X}_{\text{pre}}, \boldsymbol{\theta}_{\text{pre}})$,

$$\hat{\mathbf{L}}_{\text{nov}} = \text{CNN}_{\text{nov}}(\mathbf{X}_{\text{nov}}, \boldsymbol{\theta}_{\text{nov}}) \tag{7.5a, b}$$

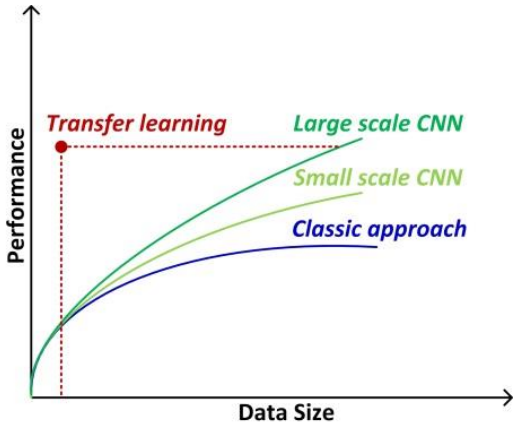


Figure 7.4 Learning methods: data size vs. performance.

CNN operator denotes the mapping of a convolutional neural network given parameters $\boldsymbol{\theta}$ from input to predicted output $\hat{\mathbf{L}}$. The parameters of the previous task is trained through

$$\boldsymbol{\theta}_{\text{pre}}' = \arg\min_{\boldsymbol{\theta}_{\text{pre}}}(\mathbf{L}_{\text{pre}} - \hat{\mathbf{L}}_{\text{pre}}) = \arg\min_{\boldsymbol{\theta}_{\text{pre}}}(\mathbf{L}_{\text{pre}} - \text{CNN}_{\text{pre}}(\mathbf{X}_{\text{pre}}, \boldsymbol{\theta}_{\text{pre}})) \tag{7.6}$$

where $\boldsymbol{\theta}_{\text{pre}}'$ stands for the parameters after training. Thereupon, the trained parameters of the first n layers can be transferred to the new task as,

$$\boldsymbol{\theta}_{\text{nov}}(1:n)' := \boldsymbol{\theta}_{\text{pre}}(1:n)' \tag{7.7}$$

The rest of the parameter can be trained using training samples from the novel task,

$$\begin{aligned}&\boldsymbol{\theta}_{\text{nov}}(1:m)' = [\boldsymbol{\theta}_{\text{nov}}(1:n)'', \boldsymbol{\theta}_{\text{nov}}(n:m)'] = \\ &\arg\min_{\boldsymbol{\theta}_{\text{nov}}(1:m)}(\mathbf{L}_{\text{nov}} - \text{CNN}_{\text{nov}}(\mathbf{X}_{\text{nov}}, [\boldsymbol{\theta}_{\text{nov}}(1:n)', \boldsymbol{\theta}_{\text{nov}}(n:m)]))\end{aligned} \tag{7.8}$$

In Equation (7.8), by setting differential learning rates, the parameters in the first n layers are fine-tuned as $\boldsymbol{\theta}_{\text{nov}}(1:n)''$ using a smaller learning rate, and the parameters in the last $(m-n)$ layers are trained from scratch as $\boldsymbol{\theta}_{\text{nov}}(n:m)'$. The phrase "differential learning rates" refers to different learning rates for different parts of the network during our training. In general, the transferred layers (i.e., the first n layers) are pre-trained to detect and extract generic features of inputs which are less sensitivity to the domain of application. Therefore, the learning rate for the transferred layers is usually very small. In an extreme case where the learning rate for the transferred layers is zero, the parameters in the first n layers transferred are left frozen.

Therefore, the CNN used for the novel task for future fault classification and diagnosis can be represented as,

$$\text{CNN}_{\text{nov}}(\mathbf{X}_{\text{nov}}, [\boldsymbol{\theta}_{\text{nov}}(1:n)'', \boldsymbol{\theta}_{nov}(n:m)']) \tag{7.9}$$

where the parameters in the first n layers are first transferred from a previous task. Meanwhile, as the last $(m-n)$ layers are trained using the training dataset of the novel task, the first n layers are fine-tuned for better results.

$$\boldsymbol{\theta}_{\text{nov}}' = [\boldsymbol{\theta}_{\text{nov}}(1:n)'', \boldsymbol{\theta}_{\text{nov}}(n:m)'] \tag{7.10}$$

Transfer learning becomes possible and promising because, as has been discovered by recent studies, the layers at the convolutional stages (convolutional layers, ReLU layers and pooling layers) of the convolutional neural network trained on large dataset indeed extract general features of inputs, while the layers of fully connected stages (fully connected layers, softmax layers, classification layers) are more

specific to task (Zeiler and Fergus, 2013; Abdeljaber et al, 2017). Therefore, the *n* layers transferred to the new task as a whole can be regarded as a well-trained feature extraction tool towards similar tasks and the last few layers serve as a classifier to be trained. Even with substantial training data, initializing with transferred parameters can improve the performance in general (Saravanan and Ramachandran, 2010).

In this research, transfer learning is implemented to gearbox fault diagnosis. The CNN is well-trained in terms of pulling characteristics from images. As illustrated in Figure 7.5, the parameters in the convolutional stage, i.e., the parameters used in the convolution filter, the ReLU operator and the max pooling operator are transferred to the fault diagnosis task. The parameters used in the fully connected layer and the softmax layers are trained subsequently using a small amount of training data generated from gear fault experiments.

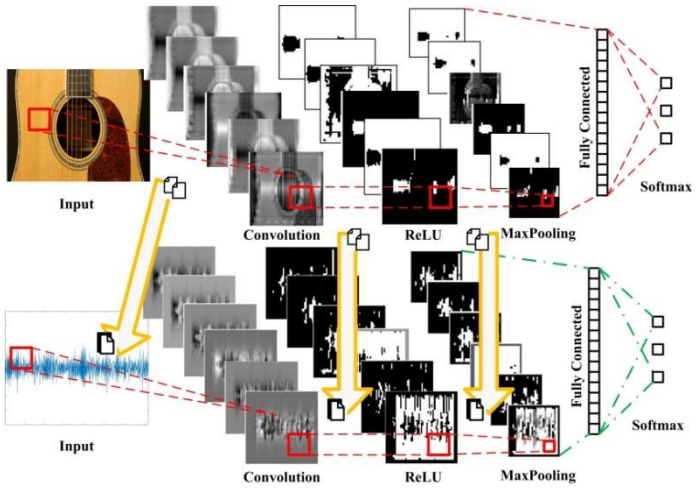


Figure 7.5 Illustration of transfer learning.

7.2.3 Proposed architecture

In this sub-section we present the proposed architecture. In gear fault diagnosis, vibration responses are recorded using accelerometers during gearbox operation. The time-domain vibration signals can then be represented directly by 2D grey-scale/true-color images (as shown in Figure 7.5) which serve as inputs of the deep CNN. More details on image representation of time-domain data will be provided in Section 7.3.1. The deep CNN adopted as the base architecture in this study was originally proposed by Krizhevsky et al (Krizhevsky et al, 2012) which is essentially composed of five convolutional stages and

three fully connected stages (Figure 7.6). This base architecture showed its extraordinary performance in Large Scale Visual Recognition Challenge 2012 (ILSVRC2012) and has since been repurposed for other learning tasks (Shie et al, 2015).

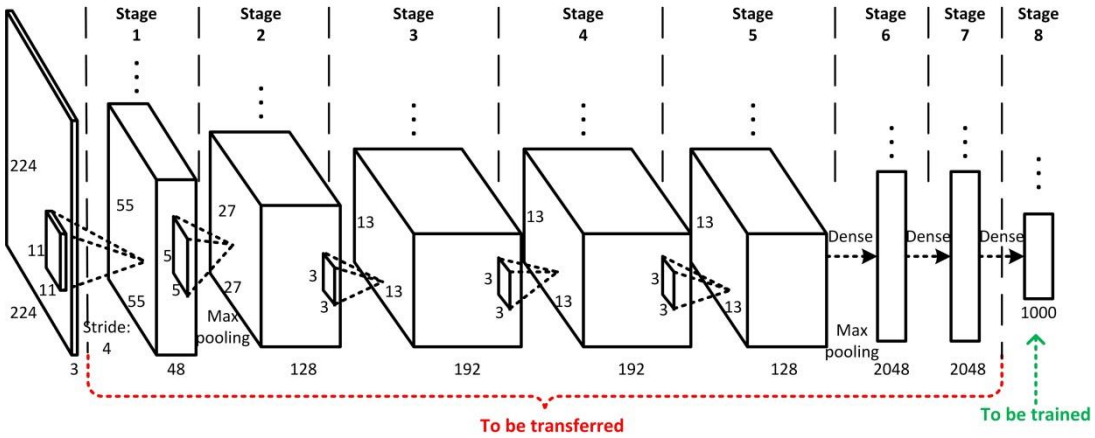


Figure 7.6 Illustration of the transfer learning architecture

In the base architecture, the parameters are trained using approximately 1.2 million human/software labeled 3D true-color nature images from ImageNet Large Scale Visual Recognition Challenge 2010 (http://www.image-net.org/challenges/LSVRC/2010/). The trained parameters in the first five stages are well-polished in characterizing high-level abstractions of the input image and thus have the potential to be used for other tasks with image inputs. Meanwhile, the last three stages are trained to nonlinearly combine the high-level features. Although the images of vibration signals may look different from the images used to train the original CNN, useful features can be extracted in a similar manner as long as the CNN adopted is capable of identifying high-level abstractions (Yosinski et al, 2014). Stage 8 of the original architecture is configured for 1000 classes in the previous image classification task. Therefore, the first seven stages of the base architecture can be possibly transferred to facilitate gear fault diagnosis. As discussed in Section 7.2.2, the first seven stages indeed serve as a general well-trained tool for automatic feature extraction. The more stages and layers used, the higher level of features can be obtained. The final stage is left to be trained as a classifier using the experimental data specific to the fault diagnosis task. As specified in Table 7.1, a total of 24 layers are used in the proposed architecture; the parameters and specifications used in the first 21 layers can be transferred from the base architecture.

Table 7.1 Specifications of the proposed architecture

Stage	Layer	Name	Specifications
1 (transferred)	1	Convolutional	11×11×96
	2	ReLU	N/A
	3	Normalization	5 channels/element
	4	Max pooling	3×3
2 (transferred)	5	Convolutional	5×5×256
	6	ReLU	N/A
	7	Normalization	5 channels/element
	8	Max pooling	3×3
3 (transferred)	9	Convolutional	3×3×384
	10	ReLU	N/A
4 (transferred)	11	Convolutional	3×3×384
	12	ReLU	N/A
5 (transferred)	13	Convolutional	3×3×256
	14	ReLU	N/A
	15	Max pooling	3×3
6 (transferred)	16	Fully connected	4096
	17	ReLU	N/A
	18	Dropout	50%
7 (transferred)	19	Fully connected	4096
	20	ReLU	N/A
	21	Dropout	50%
8 (to be trained)	22	Fully connected	9
	23	Softmax	N/A
	24	Classification	Cross entropy

We observe Table 7.1. Overfitting of the learning model is essentially controlled by the max pooling layers in Stages 1, 2, and 5, and the dropout layers in Stages 6 and 7. As explained in Section 7.2.1, a max pooling layer not only makes the network less sensitive to location changes of a feature but also reduces the size of parameters. Therefore, max pooling can reduce computational burden and control overfitting. In our architecture, dropout layers are employed after the ReLU layers in Stages 6 and 7. Because a fully connected layer possesses a large number of parameters, it is prone to overfitting. A simple and effective way to prevent from overfitting is dropout. In our study, individual nodes are

"dropped out of" (temporarily removed from) the net with probability 50% as suggested in (Srivastava et al, 2014). Dropout can be interpreted as a stochastic regularization technique which not only decreases overfitting by avoiding training all nodes, but also significantly improves training efficiency.

The loss function used is the cross-entropy function given as follows,

$$E(\boldsymbol{\theta}) = -\hat{\mathbf{L}} \ln\left(CNN(\mathbf{X}, \boldsymbol{\theta})\right) + \gamma \left\|\boldsymbol{\theta}\right\|_2 = -\hat{\mathbf{L}} \ln \mathbf{L} + \gamma \left\|\boldsymbol{\theta}\right\|_2 \tag{7.11}$$

where $\left\|\boldsymbol{\theta}\right\|_2$ is a l_2 normalization term which also contributes to preventing the network from overfitting. Equation (7.11) quantifies the difference between correct output labels and predicted labels. And the loss is then back-propagated to update the parameters using the stochastic gradient descent (SGD) method (Sutskever et al, 2013) given as,

$$\boldsymbol{\theta}_{i+1} = \boldsymbol{\theta}_i - \alpha \nabla E(\boldsymbol{\theta}_i) + \beta(\boldsymbol{\theta}_i - \boldsymbol{\theta}_{i-1}) \tag{7.12}$$

where α is the learning rate, i is the number of iteration, and β stands for the contribution of previous gradient step. While classical SGD and momentum SGD are frequently adopted in training CNNs for their simplicity and efficiency, other techniques, such as AdaGrad, AdaDelta or Adam (Kingma et al, 2014) can also be applied to carry out optimization of Equation (7.11). The transferability of the base architecture and the performance of the proposed architecture for gear fault diagnosis will be investigated in the next section.

7.3 Gear fault diagnosis implementation and demonstration

7.3.1 Data acquisition

Many types of faults and failure modes can occur to gear transmission in various machinery systems. Vibration signals collected from such a system are usually used to reveal its health condition. In this research, experimental data are collected from a benchmark two-stage gearbox with replaceable gears as shown in Figure 7.7. The gear speed is controlled by a motor. The torque is supplied by a magnetic brake which can be adjusted by changing its input voltage. A 32-tooth pinion and an 80-tooth gear are installed on the first stage input shaft. The second stage consists of a 48-tooth pinion and 64-tooth gear. The input shaft speed is measured by a tachometer, and gear vibration signals are measured by an

accelerometer. The signals are recorded through a dSPACE system (DS1006 processor board, dSPACE Inc.) with sampling frequency of 20 KHz. As shown in Figure 8, nine different gear conditions are introduced to the pinion on the input shaft, including healthy condition, missing tooth, root crack, spalling, and chipping tip with five different levels of severity. The dynamic respo nses of a system involving gear mechanism are angle-periodic. In reality, while gearbox system is recorded in a fixed sampling rate, the time-domain responses are generally not time-periodic due to speed variations under load disturbance, geometric tolerance, and motor control error etc. In order to solve the non-stationary issue and eliminate the uncertainty caused by speed varying, here we apply the time synchronous averaging (TSA) approach, where the time-even signals are resampled based on the shaft speed measured by the tachometer and averaged in angular domain. As TSA converts the signals from the time-even representation to the angle-even representation, it can significantly reduce the non-coherent components in the system response. It is worth mentioning that TSA is a standard, non-biased technique that can facilitate effective pattern recognition of various datasets (Zhang and Tang, 2018).

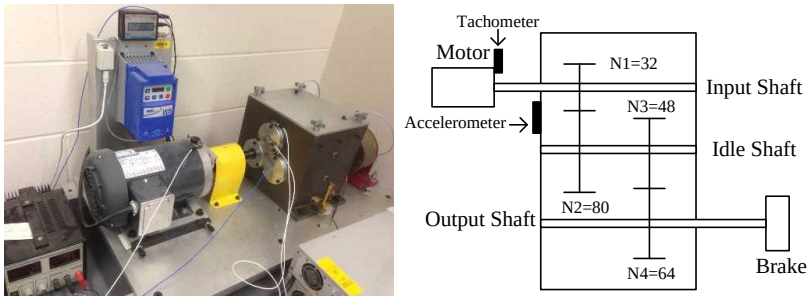
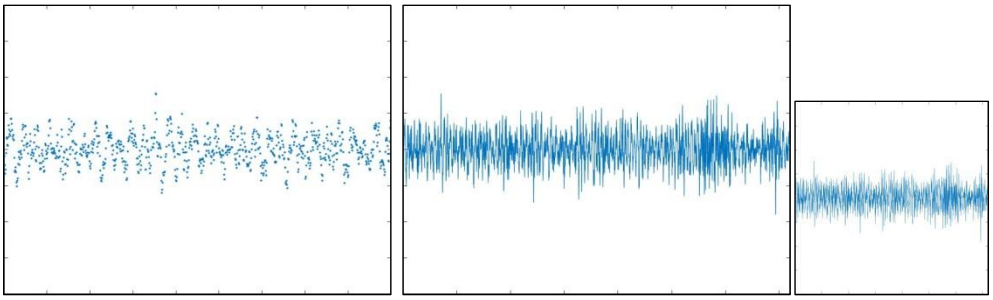


Figure 7.7 Gearbox system employed in experimental study.

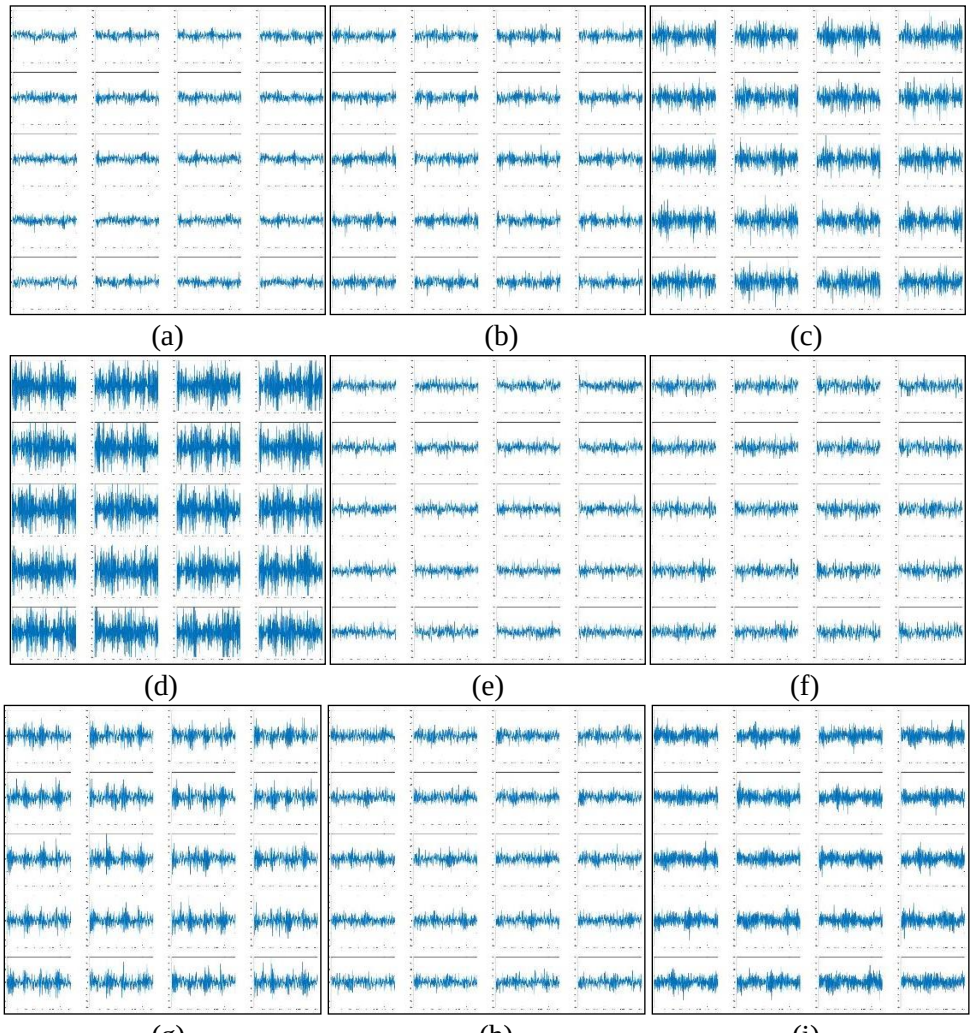
Figure 7.8 Nine pinions with different health conditions (five levels of severity for chipping tip).

To proceed, in this research we adopt a preprocessing-free approach to transform the vibration signals to images in order to discover the 2D features of raw signals. As time domain vibration signals have been cast into angle-even domain for consistency as sample points (Figure 7.9(a)), the adjacent data points are then connected in chronological sequence to generate a polyline. Figure 7.9(b) shows an example of such polyline represented in an 875×656 image generated by MATLAB plot function. The original matrix or image representation of the vibration signal is then resized to a 227×227 gray scale image using Bicubic interpolation (Prashanth et la, 2009) as shown in Figure 7.9(c). There are 51,529 pixels per image. Figure 7.10 showcases some example images generated from angle-even vibration signals. For each gear condition, 104 signals are collected using the experimental gearbox system. For each signal, 3,600 angle-even samples are recorded in the course of four gear revolutions first for the case study in Section 7.3.3, and then down-sampled to 900 angle-even points for the case study in Section 7.3.4. Figure 7.10 shows 20 example signals of each type of gear condition where the vertical axis is the acceleration of the gear (rad/s^2) and the horizontal axis corresponds to the 3,600 angel-even sampling points. All the data used in this study is made public at https://doi.org/10.6084/m9.figshare.6127874.v1.



(a) (b) (c)

Figure 7.9 Construction of input for transfer leaning. (a) 875*656 image representation of 3600 samples, (b) 875*656 image representation of the samples connected, (c) 227*227 image representation of the samples connected.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 7.10 Vibration signal examples under different gear health conditions.
(a) Healthy, (b) Missing tooth, (c) Root crack, (d) Spalling, (e) Chipping tip_5 (least severe), (f) Chipping tip_4, (g) Chipping tip_3, (h) Chipping tip_2, (i) Chipping tip_1 (most severe).

7.3.2 Setup of case illustration and comparison

In this study, in order to highlight its effectiveness, the proposed transfer learning approach is examined and compared with two contemporary approaches. As indicated, the proposed transfer learning approach does not rely on manual selection of features, and we use this approach to analyze the angle-even representation of the original time-domain signals. The first approach adopted for comparison is a three-stage (nine layers) CNN, thereafter referred to as local CNN, which consists of two convolutional stages and a fully connected stage and uses the angle-even representation of the time-domain signals as inputs. Different from the proposed approach, the local CNN will be only trained by the data generated from gearbox experiments. The specifications are the same as the stage 1, stage 2 and stage 8 given in Table 7.1. The other approach adopted for comparison is based upon manual identification/selection of features. In a recent investigation, it was recognized that the angle-frequency domain synchronous analysis (AFS) can enhance significantly fault-induced features in gearbox responses (Zhang and Tang, 2018). AFS resamples the time-domain signal into angle-domain based on the speed information collected from tachometer. The angle-domain signal is then sliced into a series of segments every four gear revolutions. Subsequently, angle-frequency analysis based on short time Fourier Transform is carried out on each segment of the angle-domain signal. The resultant spectrogram coefficients are then averaged to remove the noise and non-coherent components. As such, the features related to the gear health conditions are highly enhanced and a feature extraction technique, i.e. Principal Component Analysis, is employed to reduce the dimensionality. In this research, these low-dimensional data extracted by AFS are imported into support vector machine (SVM) for fault classification.

For the proposed transfer learning approach and the locally-trained CNN approach (local CNN), mini-batch size is set to 5, and 15 epochs are conducted meaning the training datasets are used to train the neural net 15 times throughout. The learning rate α is set to be $1e^{-4}$ and $1e^{-2}$ for transferred layers and non-transferred layers, respectively, following the suggestion in (Krizhevsky et al, 2012). The momentum β in Equation (7.12) is set to 0.9 for transfer learning and 0.5 for local CNN. For the SVM

approach based on manual feature selection, Gaussian kernel is adopted. In the next two sub-sections, the relative performance of the three approaches is highlighted as we change the sampling frequency as well as the size of the training dataset, i.e., the portion of measured gear vibration signals used for training.

Neural networks are inherently parallel algorithms. Therefore, graphical processing units (GPUs) are frequently adopted as the execution environment to take advantage of the parallelism natural of CNNs and expedite the classification process. In this research, both CNNs are trained and implemented using a single CUDA-enabled NVIDIA Quadro M2000 GPU, while AFS-SVM approach is facilitated based on an Intel Xenon E5-2640 v4 CPU.

7.3.3 Case 1 – 3,600 sampling points with varying training data size

As mentioned in Section 7.3.1, 104 vibration signals are generated for each gear condition. In the case studies, a portion of the signals are randomly selected as training data while the rest serves as validation data. To demonstrate the performance of the proposed approach towards various data sizes, the size of the training dataset ranges from 80% (83 training data per condition, 83×9 data in total) to 2% (2 training data per condition, 2×9 data in total) of all the 104 signals for each health condition.

Table 7.2 shows the classification results where the mean accuracy is the average of five training attempts. The classification accuracy is the ratio of the correctly classified validation data to the total validation dataset. As illustrated in Figure 7.11, the proposed transfer learning approach has the best classification accuracy for all types of data size. Even when only five vibration signals per condition are selected for training, the proposed approach is able to achieve an excellent 94.90% classification accuracy, which further increases to 99%-100% when 10% and more training data are used. On the other hand, while the performance of AFS-SVM reaches the plateau (showing only minimal increments) after 20% date is used for training, the classification accuracy of local CNN gradually increases with data size from 27.99% to 97.57% and surpasses AFS-SVM eventually when 80% data is used for training, indicating the significance of the size of training data in order to properly train a neural network. Although the data size greatly affects the performance of a CNN in the general sense, the proposed transfer learning architecture

exhibits very high classification accuracy because only one fully connected stage needs to be trained locally, which notably lowers the standard of the data required by a CNN in terms of achieving satisfactory outcome.

Table 7.2 Classification results (3,600 sampling points)

Method / Training data	Transfer learning Accuracy (%)		Local CNN Accuracy (%)		AFS-SVM Accuracy (%)	
80% (83 per condition)	100 100 100 100 100	Mean: 100	91.01 99.47 97.35 100 100	Mean: 97.57	86.72 88.62 87.80 87.26 86.99	Mean: 87.48
60% (62 per condition)	100 100 100 100 100	Mean: 100	90.48 97.62 58.99 88.89 67.72	Mean: 80.74	87.30 87.83 88.62 87.04 87.83	Mean: 87.72
40% (42 per condition)	100 100 100 100 100	Mean: 100	88.89 98.39 44.44 62.72 83.69	Mean: 76.63	86.74 86.38 85.84 87.99 86.38	Mean: 86.67
20% (21 per condition)	100 100 100 99.60 100	Mean: 99.92	61.31 72.56 85.41 70.41 58.77	Mean: 69.69	86.48 86.08 85.01 86.35 87.28	Mean: 86.24
10% (10 per condition)	99.88 98.23 99.88 99.29 99.76	Mean: 99.41	64.07 57.09 55.56 44.56 57.80	Mean: 55.82	80.97 86.17 78.84 86.29 86.88	Mean: 83.83
5% (5 per condition)	99.55 97.19 80.02 98.09 99.66	Mean: 94.90	65.54 37.71 31.99 28.17 57.13	Mean: 44.11	75.31 84.85 81.14 73.29 84.85	Mean: 79.89
2% (2 per condition)	76.80 73.31 69.39 73.42 68.19	Mean: 72.22	26.14 27.67 32.24 31.70 22.22	Mean: 27.99	61.87 73.97 41.72 69.72 64.92	Mean: 62.44

The average computational time consumed by each method is reported in Table 7.3, which contains the portions used for both training and classification. Generally speaking, deep neural networks are more time consuming in training compared to traditional approaches. The computational cost per iteration of a mini-batch back-propagation is proportional to the number of weights involved. And the overall computational time is linearly proportional to the size of the training data. As shown in Table 7.3, when the size of training data is small (2%), the transfer learning approach not only leads in accuracy, but also in computational efficiency compared to AFS-SVM.

Table 7.3 Computational time comparison (average of 5 attempts)

Method / Training data	Transfer learning (sec)	Local CNN (sec)	AFS-SVM (sec)
80%	588.467	373.283	52.156
60%	453.063	284.445	50.581
40%	311.824	198.517	48.181
20%	167.406	108.909	48.046
10%	98.872	64.747	47.998
5%	66.152	42.800	47.846
2%	42.840	28.847	47.781

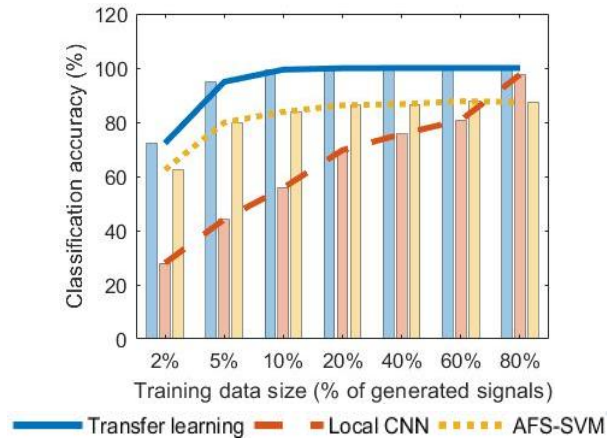


Figure 7.11 Comparison of classification results when training data size varies.

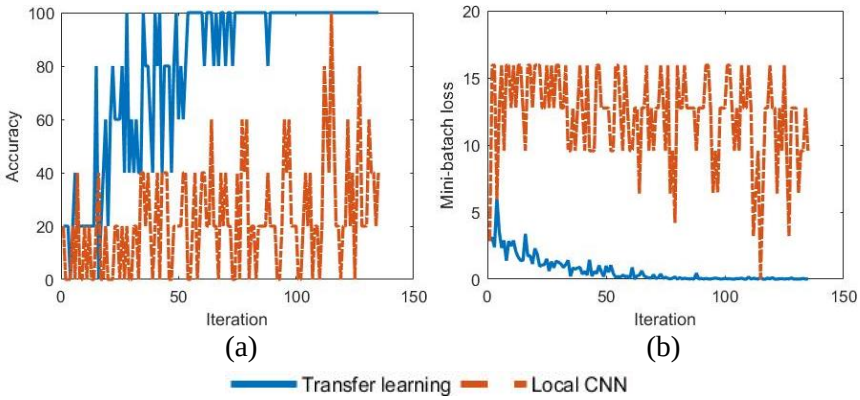
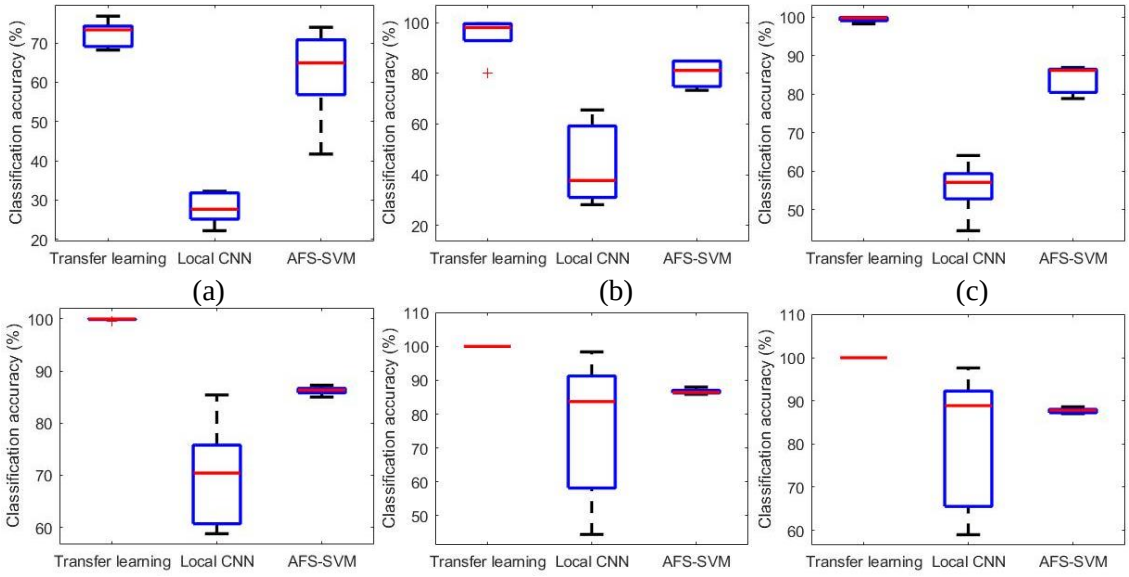
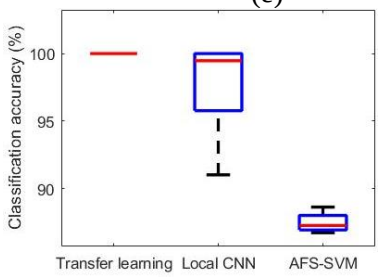


Figure 7.12 Convergent histories of transfer learning and local CNN for 5% training data. (a) Accuracy, (b) Mini-batch loss.

Figure 7.12 shows the convergent histories (mini-batch accuracy and mini-batch loss) of the proposed approach and local CNN when 5% data is used for training. As can be seen from the comparisons, transfer learning gradually converges in terms of both accuracy and loss as the training iterates while local CNN inclines to 'random walk' due to insufficient data. Compared with AFS-SVM, the proposed approach not only excels in performance, but also requires no preprocessing effort, which makes it more unbiased in feature extraction and readily applicable to other fault diagnosis practices. The proposed approach also shows satisfactory outcomes in the regard of robustness. As demonstrated in Figure 7.13, it has the smallest variance among all cases. On the other hand, the performance of the under-trained local CNN oscillates the most.



(d) (e) (f)



(g)

Figure 7.13 Comparison of box plots of classification results when training data size varies. (a) 2%, (b) 5%, (c) 10%, (d) 20%, (e) 40%, (f) 60%, (g) 80%

As mentioned in Section 7.2.3, the parameters in the first five convolutional stages of the original CNN are well-trained in characterizing high-level abstractions while the last three fully connected stages are trained to nonlinearly combine the high-level features. Hence, it is recommended to repurpose Stages 1 to 5 for novel tasks as to adaptively extract image features. Whether to transfer Stages 6 and 7 remains optional depending on the training data size. In our previous comparisons, only Stage 8 is reconstructed (from 1000 classes to 9 classes) and trained using local dataset. Here, we also compare the accuracy of the transfer learning approach when different aggregates are transferred. As shown in Table 7.4, transferring Stages 1 to 7 and transferring Stages 1 to 6 yield similar performances, which are better than transferring merely Stages 1 to 5 especially when data size is small. Recall Table 7.1. Stage 6 contains 4096 more weighting parameters, which apparently requires more training data to fine-tune even though the feature extraction passage is well-established. Moreover, transferring more layers may indeed prevent the model from overfitting because the layers transferred are already extensively trained so the generalization error is naturally reduced when the model is repurposed and only a small portion is trained by a different set of data.

Table 7.4 Classification results of transfer learnings (average of 5 attempts)

Method / Training data	Transfer learning (Stages 1-7) Accuracy (%)	Transfer learning (Stages 1-6) Accuracy (%)	Transfer learning (Stages 1-5) Accuracy (%)

80%	100	100	100
60%	100	100	100
40%	100	100	99.97
20%	99.92	99.87	99.06
10%	99.41	99.28	72.50
5%	94.90	96.30	64.54
2%	72.22	72.98	48.91

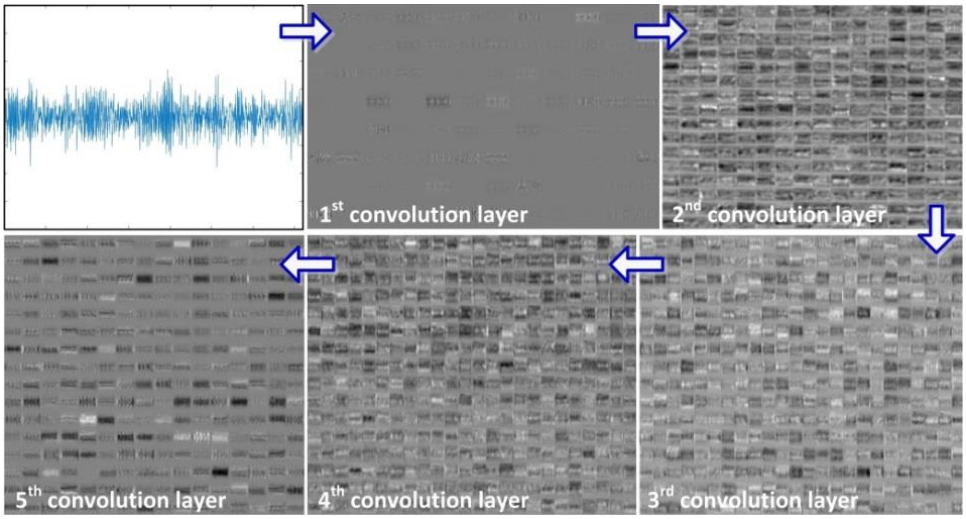
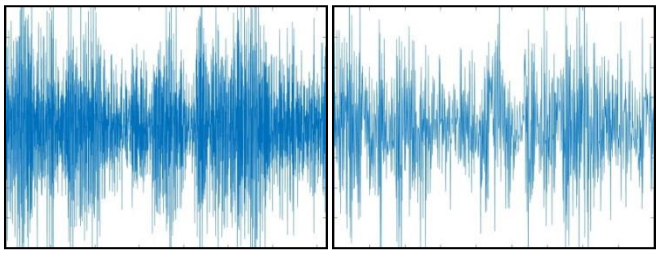


Figure 7.14 Feature maps extracted by 5 convolution layers of the proposed transfer learning approach.

As discussed in Section 7.2.2 and Section 7.2.3, the transferred stages of the proposed architecture tend to extract the high-level abstract features of the input that cannot be recognized otherwise, even if the input is different from that of the previous task. Figure 7.14 gives an example of such procedure by showing the feature maps generated in each convolutional layer by the proposed architecture when it is used to classify a gearbox vibration signal. It is seen that the abstraction level of the input image continuously escalates from the 1st feature map to the 5th feature map. In general, the number of convolutional stages equipped is correlated with the level of abstraction the features can be represented in CNNs. As demonstrated in this case study, the base architecture is indeed transferable towards gear fault diagnosis tasks and the proposed approach performs well with raw image signal inputs, which indicates the transferred layers constructed in this study are generally applicable to represent useful features of an input image in high-level abstraction.

7.3.4 Case 2 – 900 sampling points with varying training data size

In Case 1, each vibration signal is composed of 3,600 angel-even data points in the course of 4 gear revolutions. In some practical fault diagnosis systems, however, the sampling rate may be lower, which means that some features could have been lost. To take this factor into consideration and further examine the approach, we now down-sample the original vibration signals to 900 angel-even data points (Figure 7.15) and apply the same three methods for classification.



(a) (b)

Figure 15 Vibration signal of a spalling gear.
(a) 3,600 sampling points, (b) 900 sampling points

Table 7.5 lists the comparison of the classification results of the three methods with different training data sizes. Similar to Case 1, the proposed transfer learning approach is the best performer. Figure 7.16 illustrates the classification results before and after down-sampling. While lowering the sampling rate deteriorates the overall performance of all approaches, each method exhibits the similar trend as seen in Section 7.3.3. For transfer learning, it starts with 60.11% classification accuracy and reaches 95.88% when only 20% of data is used as training data whilst the accuracies of local CNN and AFS-SVM are 43.56% and 70.07%. Local CNN performs better than AFS-SVM when 80% data is used for training. Unlike AFS-SVM, the performance of local CNN can be largely improved if significantly more training data is incorporated because the parameters of lower stages can be learned from scratch. Eventually, the performance of local CNN could reach that of the transfer learning approach. Nevertheless, for cases with limited data, the proposed transfer learning approach has an extensive performance margin compared to local CNN or other preprocessing-based shallow learning methods such as AFS-SVM. Even with

ample training data, initializing with transferred parameters can improve the classification accuracy in general. Moreover, the proposed approach requires no preprocessing. Similar to Case 1 in Section 7.3.3, the proposed approach is very robust especially when 40% or more data is used for training (Figure 7.17).

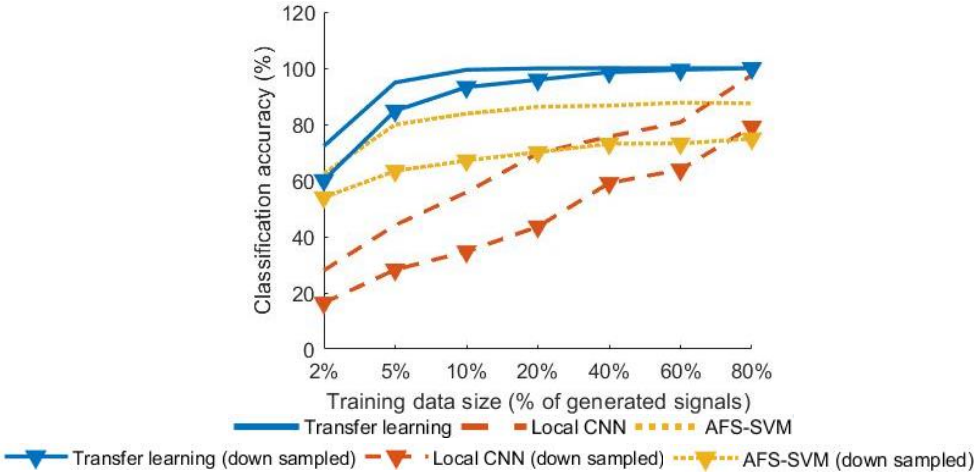


Figure 16 Classification results of the three methods after down sampling.

Table 7.5 Classification results (900 sampling points)

Method / Training data	Transfer learning Accuracy (%)		Local CNN Accuracy (%)		AFS-SVM Accuracy (%)	
80% (83 per condition)	100 100 100 100 100	Mean: 100	85.26 65.66 71.32 80.89 92.53	Mean: 79.13	74.07 74.60 75.13 76.72 74.07	Mean: 74.92
60% (62 per condition)	100 99.21 99.74 98.68 99.47	Mean: 99.42	77.25 57.67 63.76 72.22 48.41	Mean: 63.86	75.40 74.34 71.16 74.07 70.90	Mean: 73.17
40% (42 per condition)	99.10 99.10 98.92 98.92 96.77	Mean: 98.56	62.90 74.91 56.63 38.35 62.72	Mean: 59.10	74.19 72.94 72.58 73.66 72.22	Mean: 73.12
20% (21 per condition)	94.91 95.72 92.77 98.80 97.19	Mean: 95.88	34.27 40.56 44.44 44.71 53.82	Mean: 43.56	70.15 72.69 68.41 69.21 69.88	Mean: 70.07
10%	94.68	Mean:	27.78	Mean:	68.20	Mean:

(10 per condition)	93.38 90.07 92.08 95.98	93.24	39.83 46.57 17.97 41.37	34.70	68.68 65.96 66.78 65.96	67.12
5% (5 per condition)	70.73 86.65 89.12 90.24 87.43	Mean: 84.83	24.88 15.80 23.20 33.40 44.05	Mean: 28.27	64.42 62.96 63.86 65.66 60.27	Mean: 63.43
2% (2 per condition)	55.99 59.91 58.06 61.11 65.47	Mean: 60.11	21.90 17.43 22.22 9.59 11.11	Mean: 16.45	47.93 57.30 58.06 51.53 55.12	Mean: 53.99

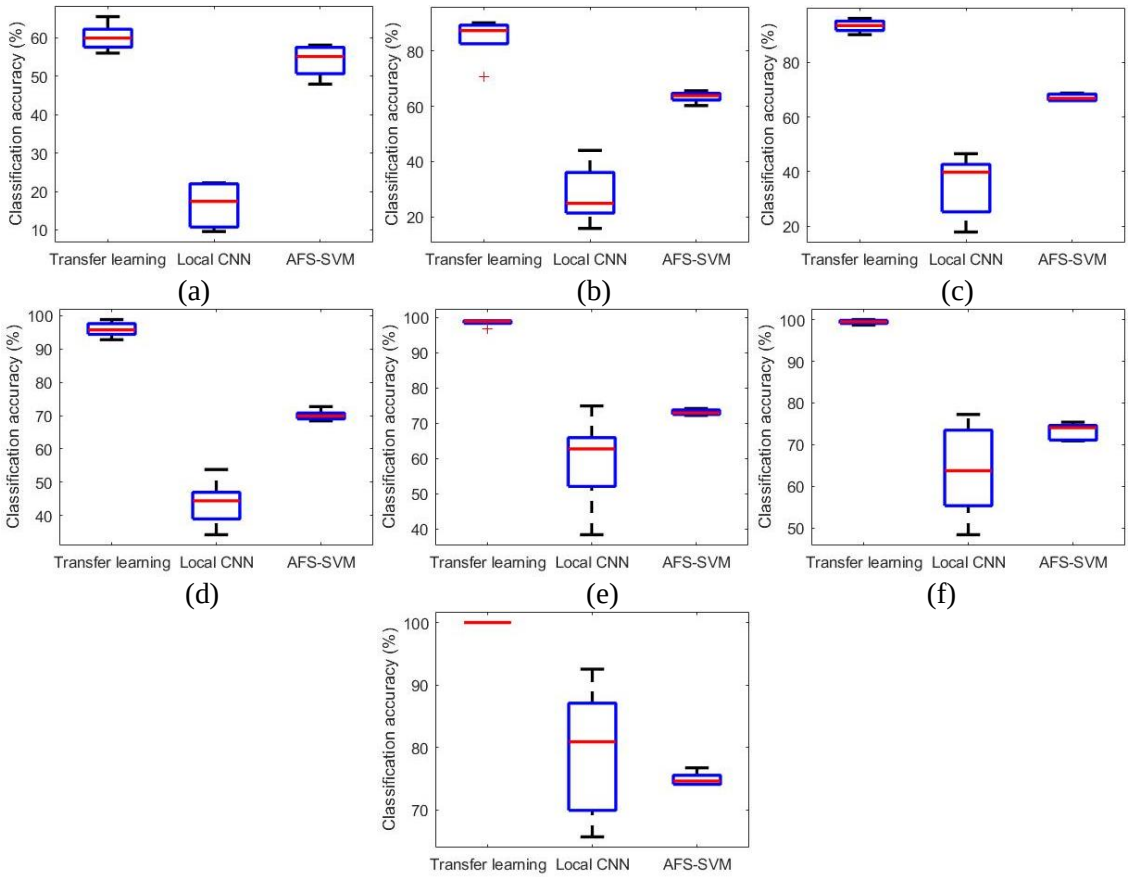


Figure 7.17 Box plots of classification results of the three methods after down sampling. (a) 2%, (b) 5%, (c) 10%, (d) 20%, (e) 40%, (f) 60%, (g) 80%

7.4 Concluding remarks

In this research, a deep convolutional neural network-based transfer learning approach is developed for deep feature extraction and applied to gear fault diagnosis. This proposed approach does not require manual feature extraction, and can be effective even with a small set of training data. Experimental studies are conducted using preprocessing free raw vibration data towards gear fault diagnose. The performance of the proposed approach is highlighted through varying the size of training data. The classification accuracies of the proposed approach outperform those of other methods such as locally trained convolutional neural network and angle-frequency analysis-based support vector machine by as much as 50%. The achieved accuracy indicates that the proposed approach is not only viable and robust, but also has the potential to be applied to fault diagnosis of other systems.

Chapter 8. Conclusion

Optimal design and identification are critical in the performance, operation and maintenance of engineering structures and systems. While the recently rapid advancements in sensing devices and computational power have caused paradigm shift in engineering analyses that emphasizes more on data driven and sampling-based approaches. The necessity of developing rapid, reliable and robust methodologies becomes important and timely. In this dissertation, data-assisted optimization and machine learning techniques are developed and employed to multi-facet engineering tasks from system design to identification. The feasibility and performance has been validated by systematic case studies. The key contributions is highlighted as follows,

In this study, a new, systematic framework to tackle the type of configuration optimization problems with multiple hard constraints is developed. The novelty of the newly developed algorithm lies in the newly designed re-seed scheme which enables the algorithm to solve the configuration optimization problem as a multi-objective optimization problem much more efficiently than existing algorithms.

We also proposed a new Focal Any-Angle A* approach. The new approach preserves the optimality advantage of visibility graph-based methods, and reduces the number of evaluations needed for path-finding compared to both gird-based (A* on Grids, Theta*) and visibility graph-based (A* on Visibility Graphs) methods. Eventually, the developed algorithm is applied to piping network design problems where optimal paths are obtained under reasonable computational time.

Based on the first study in Chapter 2, we advance the state-of-the-art in Multi-Objective Simulated Annealing (MOSA) by incorporating hyper-heuristic systematically to improve both the generality and performance. A reinforcement learning hyper-heuristic inspired by probability matching is developed, which consists of a selection strategy and a credit assignment strategy.

Next, a structural health monitoring problem is formed as a multi-objective optimization problem that can be effectively solved by a deterministic global approach originated from the Dividing RECTangles (DIRECT) algorithm. A proper choice of sparse regularization under the multi-objective formulation is

introduced that entertains the potential of yielding a *small* set of solutions that fits better the true damage scenario.

A new methodology of fault identification using piezoelectric impedance/admittance sensing is also developed. In order to balance between solution convergence and diversity, we establish an ε-dominance enabled many-objective simulated annealing algorithm based on Gaussian process regression. Subsequently, inspired by concepts in social statistics, i.e., voting power and majority voting is employed to evaluate quality of the solutions obtained. As a combination of many-objective optimization and voting score calculation, our proposed many-objective evaluation approach is able to distinguish the solutions that could accurately indicate the health condition of the structure and ultimately provide guidance for further examination.

A deep neural network-based transfer learning approach utilizing limited time-domain data for gearbox fault diagnosis is presented. One-dimensional time-domain data of vibration responses related to gear fault patterns are converted into graphical images as input. The approach inherits the non-biased nature of neural networks that can avoid the manual selection of features. Meanwhile, the issue of limited data is overcome by formulating a new neural network architecture based on transfer learning. With this new architecture, highly accurate gear fault diagnosis can be achieved using limited time-domain data directly without involving any subjective preprocessing techniques to assist feature extraction.

Bibliography

Annamdas, V.G.M., and Radhika, M.A., 2013, "Electromechanical impedance of piezoelectric transducers for monitoring metallic and non-metallic structures: A review of wired, wireless and energy-harvesting methods," Journal of Intelligent Material Systems and Structures, 24(9), pp.1021-1042.

Al-Dujaili, A. and Suresh, S., 2016, "Dividing rectangles attack multi-objective optimization," In Evolutionary Computation (CEC) 2016 IEEE Congress on (pp. 3606-3613).

Amanatides, J., A. Woo, "A fast voxel traversal algorithm for ray tracing," Eurographics, Vol. 87(3), pp. 3-10, 1987.

Antunes, C.H., Lima, P., Oliveira, E. and Pires, D.F., 2011. A multi-objective simulated annealing approach to reactive power compensation. Engineering Optimization, 43(10), pp.1063-1077.

Aiello, G., Enea, M. and Galante, G., 2006, "A multi-objective approach to facility layout problem by genetic search algorithm and Electre method," Robotics and Computer-Integrated Manufacturing, 22(5), pp.447-455.

Algfoor, Z.A., M.S. Sunar, and H. Kolivand. "A comprehensive study on pathfinding techniques for robotics and video games," International Journal of Computer Games Technology, vol. 2015(7), 2015.

Abdeljaber, O., Avci, O., Kiranyaz, S., Gabbouj, M. and Inman, D.J., 2017. Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks. Journal of Sound and Vibration, 388, pp.154-170.

Bandyopadhyay, S., Pal, S.K., Aruna, B., 2004, "Multiobjective GAs, quantitative indices, and pattern classification," IEEE Transactions on Systems Man, and Cybernetics, Part B: Cybernetics, 34(5): 2088-2099.

Bandyopadhyay, S., Saha, S., Maulik U. and Deb K., 2008, "A simulated annealing-based multiobjective optimization algorithm: AMOSA," IEEE Transactions on Evolutionary Computation, vol. 12, no.3.

Balafas, K., Kiremidjian, A.S. and Rajagopal, R., 2018. The wavelet transform as a Gaussian process for damage detection. Structural Control and Health Monitoring, V25(2), pp. e2087.

Baydar, N. and Ball, A., 2001. A comparative study of acoustic and vibration signals in detection of gear failures using Wigner–Ville distribution. Mechanical systems and signal processing, 15(6), pp.1091-1107.

Bartelmus, W. and Zimroz, R., 2009. Vibration condition monitoring of planetary gearbox under varying external load. Mechanical Systems and Signal Processing, 23(1), pp.246-257.

Bai, R., Blazewicz, J., Burke, E.K., Kendall, G. and McCollum, B., 2012. A simulated annealing hyper-heuristic methodology for flexible decision support. 4OR: A Quarterly Journal of Operations Research, 10(1), pp.43-66.

Barthorpe, R.J., Manson, G. and Worden, K., 2017, On multi-site damage identification using single-site training data. Journal of Sound and Vibration, 409, pp.43-64.

Beyer, H.G., 1997, "An alternative explanation for the manner in which genetic algorithms operate," BioSystems, 41(1): 1-15.

Begambre, O. and Laier, J.E., 2009, "A hybrid Particle Swarm Optimization–Simplex algorithm (PSOS) for structural damage identification," Advances in Engineering Software, 40(9), pp.883-891.

Breiman, L., 2001. Random forests. Machine learning, V45(1), pp.5-32.

Burke, E.K., McCollum, B., Meisels, A., Petrovic, S. and Qu, R., 2007. A graph-based hyper-heuristic for educational timetabling problems. European Journal of Operational Research, 176(1), pp.177-192.

Burke, E.K., Hyde, M.R., Kendall, G., Ochoa, G., Ozcan, E. and Woodward, J.R., 2009. Exploring hyper-heuristic methodologies with genetic programming. In Computational intelligence (pp. 177-201). Springer Berlin Heidelberg.

Burke, E.K., Gendreau, M., Hyde, M., Kendall, G., Ochoa, G., Özcan, E. and Qu, R., 2013. Hyper-heuristics: A survey of the state of the art. Journal of the Operational Research Society, 64(12), pp.1695-1724.

Botea, A., M. Müller, and J. Schaeffer, "Near optimal hierarchical path-finding," Journal of game development, vol. 1(1), pp. 7-28, 2004.

Bortfeldt, A, Wäscher, G, 2013, "Constraints in container loading -- A state-of-the-art review," European Journal of Operational Research, 229(1): 1-20.

Bresenham, J.E., "Algorithm for computer control of a digital plotter," IBM Systems journal, vol. 4(1), pp. 25-30, 1965.

Cawley, P., and F. Simonetti, 2005, "Structural health monitoring using guided waves–potential and challenges," Proceedings of the 5th International Conference on Structural Health Monitoring, DEStech Publications, Inc.

Candès, E.J., Romberg, J. and Tao, T., 2006, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," IEEE Transactions on information theory, 52(2), pp.489-509.

Cagan, J., Degentesh, D., Yin, S., 1998, "A simulated annealing-based algorithm using hierarchical models for general three-dimensional component layout," Computer-Aided Design, 30(10): 781-790.

Cao, P., Fan, Z., Gao, R., and Tang, J., 2016, "Complex Housing: Modelling and Optimization Using an Improved Multi-Objective Simulated Annealing Algorithm," Proceedings of ASME IDETC, IDETC2016-60563.

Cao, P., Z. Fan, R. Gao, and J. Tang, "A framework of a fast any-angle path finding algorithm on visibility graphs based on A* for plumbing design," In Flexible Automation (ISFA), International Symposium on (pp. 333-339). IEEE, 2016.

Cao, P., Yoo, D., Shuai, Q., and Tang, J., 2017, "Structural damage identification with multi-objective DIRECT algorithm using natural frequencies and single mode shape," Proceedings of SPIE, Smart Structures / NDE, V10170.

Cao, P., Fan, Z., Gao, R. and Tang, J., 2017. A manufacturing oriented single point search hyper-heuristic scheme for multi-objective optimization. In Proceedings of ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (pp. V02BT03A031-V02BT03A031). American Society of Mechanical Engineers.

Cao, P., Shuai, Q. and Tang, J., 2018a. A Multi-Objective DIRECT Algorithm Toward Structural Damage Identification With Limited Dynamic Response Information. Journal of Nondestructive Evaluation, Diagnostics and Prognostics of Engineering Systems, 1(2), p.021004.

Cao, P., Qi, S. and Tang, J., 2018b. Structural damage identification using piezoelectric impedance measurement with sparse inverse analysis. Smart Materials and Structures, 27(3), p.035020.

Cao, P., Fan, Z., Gao, R.X. and Tang, J., 2018. Design for Additive Manufacturing: Optimization of Piping Network in Compact System With Enhanced Path-Finding Approach. Journal of Manufacturing Science and Engineering, 140(8), p.081013.

Chen, S.S. and Donoho, D.L., 1998, "Application of basis pursuit in spectrum estimation," In Acoustics, Speech and Signal Processing, Proceedings of the 1998 IEEE International Conference on (Vol. 3, pp. 1865-1868). IEEE.

Chen, X. and Feng, Z., 2017. Time-Frequency Analysis of Torsional Vibration Signals in Resonance Region for Planetary Gearbox Fault Diagnosis Under Variable Speed Conditions. IEEE Access, 5, pp.21918-21926.

Chaari, F., Bartelmus, W., Zimroz, R., Fakhfakh, T. and Haddar, M., 2012. Gearbox vibration signal amplitude and frequency modulation. Shock and Vibration, 19(4), pp.635-652.

Cowling, P., Kendall, G. and Soubeiga, E., 2000, August. A hyperheuristic approach to scheduling a sales summit. In International Conference on the Practice and Theory of Automated Timetabling (pp. 176-190). Springer Berlin Heidelberg.

Coello, C.C., Lamont, G.B., Van Veldhuizen, D.A., 2007, *Evolutionary algorithms for solving multi-objective problems*, Springer Science & Business Media.

Cui, R., Y. Li, and W. Yan, "Mutual information-based multi-AUV path planning for scalar field sampling using multidimensional RRT", IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 46(7), pp. 993-1004, 2016.

Czyzak, P., Jaszkiewicz, A., 1998, "Pareto simulated annealing – a metaheuristic technique for multiple-objective combinatorial optimization," Journal of Multi-Criteria Decision Analysis, 6(7): 34-47.
Davenport, M.A., Duarte, M.F., Eldar, Y.C. and Kutyniok, G., 2011, "Introduction to compressed sensing," preprint, 93(1), pp. 2.
Daniel, K., A. Nash, S. Koenig, and A. Felner, "Theta*: Any-angle path planning on grids," Journal of Artificial Intelligence Research, vol. 39, pp. 533-579, 2010.
Dang, V. H., N. D. Thang, H. H. Viet, and L. A. Tuan, "Batch-Theta* for path planning to the best goal in a goal set," Advanced Robotics, vol. 29(23), pp. 1537-1550, 2015.
de Armas, J., Miranda, G. and León, C., 2011, July. Hyperheuristic encoding scheme for multi-objective guillotine cutting problems. In Proceedings of the 13th annual conference on Genetic and evolutionary computation (pp. 1683-1690). ACM.
Deb, K, 2000, "An efficient constraint handling method for genetic algorithms," Computer Methods in Applied Mechanics and Engineering, 186(2): 311-338.
Deb, K., 2001, *Multi-objective optimization using evolutionary algorithms*, John Wiley & Sons.
Deb, K., Thiele, L., Laumanns, M., Zitzler, E., 2002, "Scalable multi-objective optimization test problems," In Congress on Evolutionary Computation, pp. 825-830. IEEE Press.
Deb, K., Pratap, A., Agarwal, S., Meyarivan, T. A. M. T., 2002, "A fast and elitist multiobjective genetic algorithm: NSGA-II," Evolutionary Computation, IEEE Transactions on, 6(2), 182-197.
Deb, K. and Jain, H., 2014. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. IEEE Transactions on Evolutionary Computation, V18(4), pp.577-601.
Duro, J.A., Saxena, D.K., Deb, K. and Zhang, Q., 2014. Machine learning based decision support for many-objective optimization problems. Neurocomputing, V146, pp.30-47.
Duvenaud, D., 2014. Automatic model construction with Gaussian processes (Doctoral dissertation, University of Cambridge).
Dijkstra, E.W., "A note on two problems in connexion with graphs," Numerische Mathematik, vol. 1(1), pp. 269-271, 1959.
Donoho, D.L., 2006, "Compressed sensing," IEEE Transactions on Information Theory, 52(4), pp.1289-1306.
Dyckhoff, H., 1990, "A typology of cutting and packing problems," European Journal of Operational Research, 44(2): 145-159.
Engrand, P.A., 1998, "Multi-objective optimization approach based on simulated annealing and its application to nuclear fuel management," Technical Report, Electricite de France.
Emmerich, M., Beume, N. and Naujoks, B., 2005, March. An EMO algorithm using the hypervolume measure as selection criterion. In International Conference on Evolutionary Multi-Criterion Optimization (pp. 62-76). Springer Berlin Heidelberg.
Fakhfakh, T., F. Chaari, M. Haddar, Numerical and experimental analysis of a gear system with teeth defects, International Journal of Advanced Manufacturing Technology, 25 (2005) 542-550.
Fang, S.E., and Perera, R., 2011. Damage identification by response surface based model updating using D-optimal design. Mechanical Systems and Signal Processing, V25(2), pp.717-733.
Farrar, C.R., and Worden, K., 2006, An introduction to structural health monitoring, Philosophical Transactions of the Royal Society A, V365, pp. 303-315.
Fadel, G. M., Wiecek, M. M., 2015, "Packing optimization of free-form objects in engineering design," *Optimized Packings with Applications*, Springer International Publishing, 37-66.
Ferguson, D., and A. Stentz, "Using interpolation to improve path planning: The Field D* algorithm," Journal of Field Robotics, vol. 23(2), pp. 79-101, 2006.
Frazier, W.E., "Metal additive manufacturing: a review," Journal of Materials Engineering and Performance, vol. 23(6), pp. 1917-1928, 2014.
Finkel, D.E., 2005, "Global optimization with the DIRECT algorithm."
Gablonsky, J.M., 2001, "Modifications of the DIRECT Algorithm."

Gao, H., Guo, X., Ouyang, H. and Han, F., 2013. Crack identification of cantilever plates based on a Kriging surrogate model. ASME Journal of Vibration and Acoustics, V135(5), p.051012.
Gao, W., Huo, L., Li, H. and Song, G., 2018. An embedded tubular PZT transducer based damage imaging method for two-dimensional concrete structures. IEEE Access, V6, pp.30100-30109.
Garey, M.R., Johnson, D.S., 1979, *Computers and intractability: a guide to the theory of NP-Completeness*, Freeman, San Francisco.
Gan M, Gopinathan N, Jia X, Williams, R.A., 2004, "Predicting packing characteristics of particles of arbitrary shapes," KONA Powder and Particle Journal, 22(0): 82-93.
Gomez, J.C. and Terashima-Marín, H., 2010, November. Approximating multi-objective hyper-heuristics for solving 2d irregular cutting stock problems. In Mexican International Conference on Artificial Intelligence (pp. 349-360). Springer Berlin Heidelberg.
Guizzo, G., Fritsche, G.M., Vergilio, S.R. and Pozo, A.T.R., 2015, July. A hyper-heuristic for the multi-objective integration and test order problem. InProceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation (pp. 1343-1350). ACM.
Goldberg, D.E., Richardson, J., 1987, "Genetic algorithms with sharing for multimodal function optimization," In Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms, Hillsdale, NJ: Lawrence Erlbaum, 41-49.
Goldberg, D. E, 1989, "Genetic Algorithm in Search, Optimization and Machine Learning, Addison." Wesley Publishing Company, Reading, MA 1.98: 9.
Goldberg, D.E., 1990. Probability matching, the magnitude of reinforcement, and classifier system bidding. Machine Learning, 5(4), pp.407-425.
González-Cruz, M.C. and Martínez, E.G.S., 2011, "An entropy-based algorithm to solve the facility layout design problem," Robotics and Computer-Integrated Manufacturing, 27(1), pp.88-100.
Grignon, P. M., Fadel, G. M., 2004, "A GA based configuration design optimization method," Journal of Mechanical Design, 126(1): 6-15.
Giagkiozis, I., Purshouse, R.C., Fleming, P.J, 2015, "An overview of population-based algorithms for multi-objective optimization," International Journal of Systems Science, 46(9): 1572-1599.
Gibson, I., D.W. Rosen, and B. Stucker, Additive manufacturing technologies, vol. 238, New York: Springer, 2010.
Harabor, D. D., A. Grastien, "An optimal any-angle pathfinding algorithm," ICAPS, Jun. 2013.
Hart, P.E., N.J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths", IEEE Transactions on Systems Science and Cybernetics, vol. 4(2), pp. 100-107, 1968.
Harley, J.B., and Moura, J.M.F., 2014, "Data-driven matched field processing for Lamb wave structural health monitoring," The Journal of the Acoustical Society of America, 135(3), pp.1231-1244.
Hershberger, J., and S. Suri. "An optimal algorithm for Euclidean shortest paths in the plane," SIAM Journal on Computing, vol. 28(6), pp. 2215-2256, 1999.
Hitomi, N. and Selva, D., 2015, August. The effect of credit definition and aggregation strategies on multi-objective hyper-heuristics. In ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference (pp. V02BT03A030-V02BT03A030). American Society of Mechanical Engineers.
Hitomi, N. and Selva, D., 2016. A Classification and Comparison of Credit Assignment Strategies in Multiobjective Adaptive Operator Selection. IEEE Transactions on Evolutionary Computation.
Hormann, K., and A. Agathos, "The point in polygon problem for arbitrary polygons," Computational Geometry, vol. 20(3), pp. 131-144, 2001.
Huang, Y., Beck, J.L. and Li, H., 2017. Bayesian system identification based on hierarchical sparse Bayesian learning and Gibbs sampling with application to structural damage assessment. Computer Methods in Applied Mechanics and Engineering, 318, pp.382-411.
Huband, S., Hingston, P., Barone, L., While, L, 2006, "A review of multiobjective test problems and a scalable test problem toolkit," IEEE Transactions on Evolutionary Computation, 10(5): 477-506.

Ishibuchi, H., Sakane, Y., Tsukamoto, N. and Nojima, Y., 2010, July. Simultaneous use of different scalarizing functions in MOEA/D. In Proceedings of the 12th annual conference on Genetic and evolutionary computation (pp. 519-526). ACM.

Ince, T., Kiranyaz, S., Eren, L., Askar, M. and Gabbouj, M., 2016. Real-time motor fault detection by 1-D convolutional neural networks. IEEE Transactions on Industrial Electronics, 63(11), pp.7067-7075.

Jain, A. K., and C.D. Richard, Algorithms for clustering data, Prentice-Hall, Inc., 1988, pp. 50-70.

Jones, D.R., Perttunen, C.D. and Stuckman, B.E., 1993, "Lipschitzian optimization without the Lipschitz constant," Journal of Optimization Theory and Applications, 79(1), pp.157-181.

Jiang, L.J., Tang, J. and Wang, K.W., 2006, "An enhanced frequency-shift-based damage identification method using tunable piezoelectric transducer circuitry," Smart Materials and Structures, 15(3), pp.799-808.

Jin, S.S. and Jung, H.J., 2016. Sequential surrogate modeling for efficient finite element model updating. Computers & Structures, V168, pp.30-45.

Jassim, Z.A., Ali, N.N., Mustapha, F. and Jalil, N.A., 2013, "A review on the vibration analysis for a damage occurrence of a cantilever beam," Engineering Failure Analysis, 31, pp.442-461.

Kang, D., Xiaoyong, Z. and Yahua, C., 2001. The vibration characteristics of typical gearbox faults and its diagnosis plan. Journal of vibration and shock, 20(3), pp.7-12.

Kaipio, J. and Somersalo, E., 2006, "Statistical and computational inverse problems," Vol. 60. Springer Science & Business Media.

Kaelbling, L.P., Littman, M.L. and Moore, A.W., 1996. Reinforcement learning: A survey. Journal of artificial intelligence research, 4, pp.237-285.

Kaveh, A. and Laknejadi, K., 2013. A new multi-swarm multi-objective optimization method for structural design. Advances in Engineering Software, 58, pp.54-69.

Kennedy, M.C. and O'Hagan, A., 2001. Bayesian calibration of computer models. Journal of the Royal Statistical Society: Series B (Statistical Methodology), V63(3), pp.425-464.

Kim, J, and Wang, K.W., 2014, "An enhanced impedance-based damage identification method using adaptive piezoelectric circuitry," Smart Materials and Structures, 23(9), 095041.

Kim, J.T. and Stubbs, N., 2003, "Crack detection in beam-type structures using frequency data," Journal of Sound and Vibration, 259(1), pp.145-160.

Kim, Y., 2014. Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.

Kingma, D.P. and Ba, J., 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P., 1983, "Optimization by Simulated Annealing," Science, New Series, 220(4598): 671-680.

Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).

Krempser, E., Fialho, Á. and Barbosa, H.J., 2012, September. Adaptive operator selection at the hyper-level. In International Conference on Parallel Problem Solving from Nature (pp. 378-387). Springer Berlin Heidelberg.

Konar, A., I.G. Chakraborty, S.J. Singh, L.C. Jain, and A.K. Nagar, "A deterministic improved Q-learning for path planning of a mobile robot," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 43(5), pp.1141-1153, 2013.

Koenig, S., and M. Likhachev, "D* Lite," in Proc. AAAI/IAAI. 2002, pp. 476-483.

Laumanns, M., Thiele, L., Deb, K. and Zitzler, E., 2002, "Combining convergence and diversity in evolutionary multiobjective optimization," Evolutionary computation, 10(3), pp.263-282.

Lam, L. and Suen, S.Y., 1997. Application of majority voting to pattern recognition: an analysis of its behavior and performance. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, 27(5), pp.553-568.

LaValle, S.M., Planning algorithms, Cambridge, UK, Cambridge University Press, 2006.

LeCun, Y., Boser, B.E., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W.E. and Jackel, L.D., 1990. Handwritten digit recognition with a back-propagation network. In Advances in neural information processing systems (pp. 396-404).

Li, K., Deb, K., Zhang, Q. and Kwong, S., 2015. An evolutionary many-objective optimization algorithm based on dominance and decomposition. IEEE Transactions on Evolutionary Computation, V19(5), pp.694-716.

Li, D.Z., Wang, W. and Ismail, F., 2015. An enhanced bispectrum technique with auxiliary frequency injection for induction motor health condition monitoring. IEEE Transactions on Instrumentation and Measurement, 64(10), pp.2679-2687.

Li, J. and Law, S.S., 2011. Substructural response reconstruction in wavelet domain. ASME Journal of Applied Mechanics, V78(4), p.041010.

Li, M., Yang, S. and Liu, X., 2016. Pareto or Non-Pareto: Bi-Criterion Evolution in Multiobjective Optimization. IEEE Transactions on Evolutionary Computation, 20(5), pp.645-665.

Li, C., Sanchez, R.V., Zurita, G., Cerrada, M., Cabrera, D. and Vásquez, R.E., 2016. Gearbox fault diagnosis based on deep random forest fusion of acoustic and vibratory signals. Mechanical Systems and Signal Processing, 76, pp.283-293.

Liu, Y., and S. Arimoto, "Path planning using a tangent graph for mobile robots among polygonal and curved obstacles communication," The International Journal of Robotics Research, vol. 11(4), pp.376-382, 1992.

Lipson, H., and M. Kurman, 2013, "Fabricated: The new world of 3D printing," John Wiley & Sons.

Lim, Y. Y., & Soh, C. K., 2014, "Towards more accurate numerical modeling of impedance based high frequency harmonic vibration," Smart Materials and Structures, 23(3), 035017.

Lim, Y. Y., & Soh, C. K., 2012, "Effect of varying axial load under fixed boundary condition on admittance signatures of electromechanical impedance technique," Journal of Intelligent Material Systems and Structures, 23(7), 815-826.

Lin, J. and Zuo, M.J., 2003. Gearbox fault diagnosis using adaptive wavelet filter. Mechanical systems and signal processing, 17(6), pp.1259-1269.

Lu, Y., Tang, J. and Luo, H., 2012. Wind turbine gearbox fault detection using multiple sensors with features level data fusion. Journal of Engineering for Gas Turbines and Power, 134(4), p.042501.

Lu, C., Xiao, S., Li, X. and Gao, L., 2016. An effective multi-objective discrete grey wolf optimizer for a real-world scheduling problem in welding production. Advances in Engineering Software, 99, pp.161-176.

Lozano-Pérez, T., and M. A. Wesley, "An algorithm for planning collision-free paths among polyhedral obstacles," Communications of the ACM, vol. 22(10), pp. 560-570, 1979.

Márquez, F.P.G., A.M. Tobias, J.M.P. Pérez, M. Papaelias, Condition monitoring of wind turbines: Techniques and methods, Renewable Energy, 46 (2012) 169-178.

Maity, D. and Tripathy, R.R., 2005, "Damage assessment of structures from changes in natural frequencies using genetic algorithm," Structural Engineering and Mechanics, 19(1), pp.21-42.

Mascareñas, D., Cattaneo, A., Theiler, J. and Farrar, C., 2013, "Compressed sensing techniques for detecting damage in structures," Structural Health Monitoring, 12(4), pp.325-338.

Maturana, J., Fialho, Á., Saubion, F., Schoenauer, M. and Sebag, M., 2009, May. Extreme compass and dynamic multi-armed bandits for adaptive operator selection. In Evolutionary Computation, 2009. CEC'09. IEEE Congress on (pp. 365-372). IEEE.

Mann, J.W., Smith, G.D., 1996, "A comparison of heuristics for telecommunications traffic routing," Modern Heuristic Search Methods, 235-254.

Maashi, M., Kendall, G. and Özcan, E., 2015. Choice function based hyper-heuristics for multi-objective optimization. Applied Soft Computing, 28, pp.312-326.

McClymont, K. and Keedwell, E.C., 2011, July. Markov chain hyper-heuristic (MCHH): an online selective hyper-heuristic for multi-objective continuous problems. In Proceedings of the 13th annual conference on Genetic and evolutionary computation (pp. 2003-2010). ACM.

Messina, A., Williams, E.J. and Contursi, T., 1998, "Structural Damage Detection by A Sensitivity and Statistical-Based Method," Journal of Sound and Vibration, 216(5), pp.791-808.
Mezura-Montes, E., Coello, C.A.C, 2006, "A survey of constraint-handling techniques based on evolutionary multiobjective optimization," Workshop paper at PPSN.
Müller, A., Schneider J.J., Schömer E., 2009, "Packing a multidisperse system of hard disks in a circular environment," Physical Review E, 79(2): 021102.
Mitra, D., Romeo, F., Vincentelli, A.S., 1986, "Convergence and finite-time behavior of simulated annealing," Advances in Applied Probability, 747-771.
Mitchell, J.S.B., D.M. David, and C.H. Papadimitriou, "The discrete geodesic problem," SIAM Journal on Computing, vol. 16(4), pp. 647-668, 1987.
Mitchell, J. S. B., and C. H. Papadimitriou, "The weighted region problem: finding shortest paths through a weighted planar subdivision," Journal of the ACM (JACM), vol. 38(1), pp. 18-73, 1991.
Michaels, J.E., and Michaels, T.E., 2007, "Guided wave signal processing and image fusion for in situ damage localization in plates," Wave Motion, 44(6), pp.482-492.
Min, J., Park, S., Yun, C.B., Lee, C.G. and Lee, C., 2012, "Impedance-based structural health monitoring incorporating neural network technique for identification of damage type and severity," Engineering Structures, 39, pp.210-220.
Nash, A., and S. Koenig, "Any-angle path planning," AI Magazine, vol. 34(4), pp. 85-107, 2013.
Nash, A.,, K. Daniel, S. Koenig and A. Felner, "Theta*: any-angle path planning on grids," in Proc. National Conference on Artificial Intelligence, vol. 22(2), Menlo Park, CA; Cambridge, MA, London, AAAI Press, MIT Press, 1999, 2007.
Nash, A.,, S. Koenig, and C. Tovey, "Lazy Theta*: Any-angle path planning and path length analysis in 3D," in Third Annual Symposium on Combinatorial Search, Aug. 2010.
Nareyek, A., 2003. Choosing search heuristics by non-stationary reinforcement learning. In Metaheuristics: Computer decision-making (pp. 523-544). Springer, Boston, MA.
Nam, D., Park, C.H., 2000, "Multiobjective simulated annealing: A comparative study to evolutionary algorithms," International Journal of Fuzzy Systems, 2(2): 87-97.
Nageshwaraniyer, S.S., Khilwani, N., Tiwari, M.K., Shankar, R. and Ben-Arieh, D., 2013, "Solving the design of distributed layout problem using forecast windows: a hybrid algorithm approach," Robotics and Computer-Integrated Manufacturing, 29(1), pp.128-138.
Neal, R.M., 2011. MCMC using Hamiltonian dynamics. Handbook of Markov Chain Monte Carlo, V2(11).
Özcan, E., Misir, M., Ochoa, G. and Burke, E.K., 2012. A Reinforcement Learning: Great-Deluge Hyper-Heuristic for Examination Timetabling. In Modeling, Analysis, and Applications in Metaheuristic Computing: Advancements and Trends (pp. 34-55). IGI Global.
Pachaud, C., Salvetat, R. and Fray, C., 1997. Crest factor and kurtosis contributions to identify defects inducing periodical impulsive forces. Mechanical systems and signal processing, 11(6), pp.903-916.
Park, G., Sohn, H., Farrar, C.R. and Inman, D.J., 2003, "Overview of piezoelectric impedance-based health monitoring and path forward," Shock and Vibration Digest, 35(6), pp.451-464.
Park, S., Lee, J., Yun, C., and Inman, D. J., 2008, Electro-mechanical impedance-based wireless structural health monitoring using PCA-data compression and k-means clustering algorithms, Journal of Intelligent Material Systems and Structures, V19, pp.509-520.
Parey, A. and Pachori, R.B., 2012. Variable cosine windowing of intrinsic mode functions: Application to gear fault diagnosis. Measurement, 45(3), pp.415-426.
Perera, R., Fang, S.E. and Ruiz, A., 2010, "Application of particle swarm optimization and genetic algorithms to multiobjective damage identification inverse problems with modelling errors," Meccanica, 45(5), pp.723-734.
Patel, A., "Amit's game programming information," International Journal of Robotics Research, 2000.
Pinedo, M., 2005, *Planning and scheduling in manufacturing and services*, Vol. 24. New York: Springer.

Prashanth, H.S., Shashidhara, H.L. and KN, B.M., 2009, December. Image scaling comparison using universal image quality index. In Advances in Computing, Control, & Telecommunication Technologies, 2009. ACT'09. International Conference on (pp. 859-863). IEEE.

Qian, S. and Chen, D., 1999. Joint time-frequency analysis. IEEE Signal Processing Magazine, 16(2), pp.52-67.

Rasmussen, C.E. and Williams, C.K., 2006. Gaussian processes for machine learning (Vol. 1). Cambridge: MIT press.

Randall, R.B., Vibration-based condition monitoring: industrial, aerospace and automotive applications, John Wiley & Sons, West Sussex, United Kingdom, 2011.

Rabin, S., "A* speed optimizations," Game Programming Gems, vol. 1, pp. 272-287, 2000.

Rabin, S., "Implementing a state machine language," AI Game Programming Wisdom, pp. 314-320, 2002.

Raad, D., Sinske, A. and van Vuuren, J., 2010. Multiobjective optimization for water distribution system design using a hyperheuristic. Journal of Water Resources Planning and Management, 136(5), pp.592-596.

Rakshit, P., A. Konar, P. Bhowmik, I. Goswami, S. Das, L.C. Jain, and A.K. Nagar, "Realization of an adaptive memetic algorithm using differential evolution and q-learning: a case study in multirobot path planning," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 43(4), pp.814-831, 2013.

Ray, T., Tai, K., Seow, K.C., 2001, "Multiobjective design optimization by an evolutionary algorithm," Engineering Optimization, 33(4), 399-424.

Ren, W.X. and Chen, H.B., 2010. Finite element model updating in structural dynamics by using the response surface method. Engineering structures, V32(8), pp.2455-2465.

Ryan, C., 1995, "Niche and species formation in genetic algorithms," Practical Handbook of Genetic Algorithms, 1: 57-74.

Saravanan, N. and Ramachandran, K.I., 2010. Incipient gear box fault diagnosis using discrete wavelet transform (DWT) for feature extraction and classification using artificial neural network (ANN). Expert Systems with Applications, 37(6), pp.4168-4181.

Sabar, N.R., Ayob, M., Qu, R. and Kendall, G., 2012. A graph coloring constructive hyper-heuristic for examination timetabling problems. Applied Intelligence, 37(1), pp.1-11.

Sachdev, S., Paredis, C. J. J., Gupta, S. K., Talukdar, S. N., 1998, "3D spatial layouts using a-teams," Proceedings of ASME 24th Design Automation Conference, DETC98/DAC-5628.

Shuai, Q., Zhou, K., Zhou, S. and Tang, J., 2017. Fault identification using piezoelectric impedance measurement and model-based intelligent inference with pre-screening. Smart Materials and Structures, 26(4), p.045007.

Seyedpoor, S. M., S. Shahbandeh, and O. Yazdanpanah, 2015, "An efficient method for structural damage detection using a differential evolution algorithm-based optimisation approach," Civil Engineering and Environmental Systems, 32(3), pp.230-250.

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R. and LeCun, Y., 2013. Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229.

Standards Committee of the IEEE Ultrasonics, Ferroelectrics, and Frequency Control Society 1987 An American National Standard: IEEE Standard on Piezoelectricity, ANSI/IEEE Std176-1987 (The Institute of Electrical and Electronics Engineers, New York.

Šišlák, David, Premysl Volf, and Michal Pechoucek. "Accelerated A* trajectory planning: Grid-based path planning comparison," in Proc. ICAPS, 2009, pp. 74-81.

Suman, B., 2004, "Study of simulated annealing based algorithms for multiobjective optimization of a constrained problem," Computers & Chemical Engineering, 28(9): 1849-1871.

Suman, B., 2005, "Study of self-stopping PDMOSA and performance measure in multiobjective optimization," Computers & Chemical Engineering, 29(5): 1131-1147.

Suman, B., Kumar P., 2006, "A survey of simulated annealing as a tool for single and multiobjective optimization." Journal of the operational research society 57.10: 1143-1160.

Suppapitnarm, A., Seffen, K.A., Parks, G.T., Clarkson, P.J, 2000, "A simulated annealing algorithm for multiobjective optimization," Engineering Optimization, 33(1): 59-85.
Smith, K.I., 2006, "A study of simulated annealing techniques for multi-objective optimisation," Thesis, University of Exeter.
Sturtevant, N. R., and R. Geisberger. "A comparison of high-level approaches for speeding up pathfinding," AIIDE, Oct. 2010.
Srinivas, N. and Deb, K., 1994, "Muiltiobjective optimization using nondominated sorting in genetic algorithms," Evolutionary computation, 2(3), pp.221-248.
Szykman, S., Cagan, J., 1995, "A simulated annealing-based approach to three-dimensional component packing," Journal of Mechanical Design, 117(2A): 308-314.
Shie, C.K., Chuang, C.H., Chou, C.N., Wu, M.H. and Chang, E.Y., 2015, August. Transfer representation learning for medical image analysis. In Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE (pp. 711-714). IEEE.
Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R., 2014. Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, 15(1), pp.1929-1958.
Sutskever, I., Martens, J., Dahl, G. and Hinton, G., 2013, February. On the importance of initialization and momentum in deep learning. In International conference on machine learning(pp. 1139-1147).
Tang, B., W. Liu, T. Song, Wind turbine fault diagnosis based on Morlet wavelet transformation and Wigner-Ville distribution, Renewable Energy, 35 (2010) 2862-2866.
Taylor, A.D. and Pacelli, A.M., 2008. Mathematics and Politics: Strategy, Voting, Power, and Proof. Springer Science & Business Media.
Tarantola, A., 2005, "Inverse problem theory and methods for model parameter estimation," Society for Industrial and Applied Mathematics.
Tanaka, M., Watanabe, H., Furukawa, Y. and Tanino, T., 1995, "GA-based decision support system for multicriteria optimization," In Systems, Man and Cybernetics, 1995. Intelligent Systems for the 21st Century, IEEE International Conference on (Vol. 2, pp. 1556-1561).
Thierens, D., 2007. Adaptive strategies for operator allocation. In Parameter Setting in Evolutionary Algorithms (pp. 77-90). Springer Berlin Heidelberg.
Thorpe, C., L. Matthies, "Path relaxation: path planning for a mobile robot," in Proc. IEEE OCEANS 1984, Sept. 1984, pp. 576-581.
Ulungu, E.L., Teghem, J.F.P.H., Fortemps, P.H., Tuyttens, D., 1999, "MOSA method: a tool for solving multiobjective combinatorial optimization problems," Journal of Multi-criteria Decision Analysis, 8(4): 221.
Van den Oord, A., Dieleman, S. and Schrauwen, B., 2013. Deep content-based music recommendation. In Advances in neural information processing systems (pp. 2643-2651).
Wang, Y. and Li, B., 2010. Multi-strategy ensemble evolutionary algorithm for dynamic multi-objective optimization. Memetic Computing, 2(1), pp.3-24.
Wang, W.X., Wang, X., Ge, X.L. and Deng, L., 2014. Multi-objective optimization model for multi-project scheduling on critical chain. Advances in Engineering Software, 68, pp.33-39.
Wang, L., Ishida, H., Hiroyasu, T. and Miki, M., 2008, "Examination of multi-objective optimization method for global search using direct and ga," In Evolutionary Computation. CEC 2008 (IEEE World Congress on Computational Intelligence). IEEE Congress on (pp. 2446-2451).
Wang, K.W. and Tang, J., 2008, Adaptive Structural Systems with Piezoelectric Transducer Circuitry, Springer Science & Business Media.
Wang, X., and Tang, J., 2010, "Damage detection using piezoelectric impedance approach with inductive circuitry," Journal of Intelligent Material Systems and Structures, 21(7), pp.667-676.
Wang, Y. and Hao, H., 2013, "Damage identification scheme based on compressive sensing," Journal of Computing in Civil Engineering, 29(2), pp. 04014037.
Wan, H.P. and Ren, W.X., 2015. A residual-based Gaussian process model framework for finite element model updating. Computers & Structures, V156, pp.149-159.

Wang, Z., Ma, J. and Zhang, L., 2017. State-of-health estimation for lithium-ion batteries based on the multi-island genetic algorithm and the Gaussian process regression. IEEE Access, V5, pp.21286-21295.

Weimer, D., Scholz-Reiter, B. and Shpitalni, M., 2016. Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection. CIRP Annals-Manufacturing Technology, 65(1), pp.417-420.

Wen, W., Z. Fan, D. Karg, W. Cheng, Rolling element bearing fault diagnosis based on multiscale general fractal features, Shock and Vibration, 2015 (2015).

Wong, C.S.Y., Al-Dujaili, A. and Sundaram, S., 2016, "Hypervolume-based DIRECT for multi-objective optimisation," In Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion (pp. 1201-1208). ACM.

Wipf, D.P. and Rao, B.D., 2004, "Sparse Bayesian learning for basis selection," IEEE Transactions on Signal processing, 52(8), pp.2153-2164.

Xia, Y. and Hao, H., 2003, "Statistical damage identification of structures with frequency changes," Journal of Sound and Vibration, 263(4), pp.853-870.

Xia, Z. and Tang, J., 2013. Characterization of dynamic response of structures with uncertainty by using Gaussian processes. ASME Journal of Vibration and Acoustics, V135(5), p.051006.

Yan, R., R.X. Gao, X. Chen, Wavelets for fault diagnosis of rotary machines: a review with applications, Signal Processing, 96 (2014) 1-15.

Yang, J., Li, S. and Xu, W., 2018. Active Learning for Visual Image Classification Method Based on Transfer Learning. IEEE Access, 6, pp.187-198.

Yang, T. and Hung, C.C., 2007, "Multiple-attribute decision making methods for plant layout design problem," Robotics and computer-integrated manufacturing, 23(1), pp.126-137.

Yang, Y., Lim, Y. Y., & Soh, C. K., 2008, "Practical issues related to the application of the electromechanical impedance technique in the structural health monitoring of civil structures: I. Experiment," Smart Materials and Structures, 17(3), 035008.

Yang, X., Guo, X., Ouyang, H. and Li, D., 2017. A Kriging model based finite element model updating method for damage detection. Applied Sciences, V7(10), p.1039.

Yap, P., N. Burch, R.C. Holte, and J. Schaeffer, "Block A*: database-driven search with applications in any-angle path-planning," In AAAI, 2011.

Yosinski, J., Clune, J., Bengio, Y. and Lipson, H., 2014. How transferable are features in deep neural networks?. In Advances in neural information processing systems (pp. 3320-3328).

Zaretalab, A., Hajipour, V., Sharifi, M., Shahriari, M.R., 2015, "A knowledge-based archive multi-objective simulated annealing algorithm to optimize series -- parallel system with choice of redundancy strategies," Computers & Industrial Engineering, 80: 33-44.

Zhang, S. and Tang, J., 2018. Integrating angle-frequency domain synchronous averaging technique with feature extraction for gear fault diagnosis. Mechanical Systems and Signal Processing, 99, pp.711-729.

Zhang, C., Sun, J.H. and Tan, K.C., 2015, October. Deep belief networks ensemble with multi-objective optimization for failure diagnosis. In Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on (pp. 32-37). IEEE.

Zhang, R., Tao, H., Wu, L. and Guan, Y., 2017. Transfer Learning with Neural Networks for Bearing Fault Diagnosis in Changing Working Conditions. IEEE Access, 5, pp.14347-14357.

Zhang, Q., and Li H., 2007, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition." Evolutionary Computation, IEEE Transactions on11.6: 712-731.

Zhang, Q., Zhou, A., Zhao, S., Suganthan, P.N., Liu, W. and Tiwari, S., 2008. Multiobjective optimization test instances for the CEC 2009 special session and competition. University of Essex, Colchester, UK and Nanyang technological University, Singapore, special session on performance assessment of multi-objective optimization algorithms, technical report, 264.

Zhou, C., "A direct tool path planning algorithm for line scanning based stereolithography," Journal of Manufacturing Science and Engineering, vol. 136(6), 061023, 2014.

Zhou, W. and Zuo, L., 2012, “Sensitivity-enhanced admittance-based structure health monitoring using a higher-order resonant circuit,” Smart Materials and Structures, 21(10), 105023.

Zhou, W., Habetler, T.G. and Harley, R.G., 2008. Bearing fault detection via stator current noise cancellation and statistical control. IEEE Transactions on Industrial Electronics, 55(12), pp.4260-4269.

Zitzler, E., & Thiele, L., 1999, “Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach,” IEEE Transactions on Evolutionary Computation, 257.

Zitzler, E. 1999. Evolutionary algorithms for multiobjective optimization: Methods and applications. Vol. 63. Ithaca: Shake.

Zeiler, M.D. and Fergus, R., 2013. Stochastic pooling for regularization of deep convolutional neural networks. arXiv preprint arXiv:1301.3557.